

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2012

Bc. Václav Kubíček

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského
inženýrství

Měření teplot ve vrtech pro tepelná čerpadla
Temperature Measurement in Borehole Heat Exchangers

2012

Bc. Václav Kubíček

Zadání diplomové práce

Student:

Bc. Václav Kubíček

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2601T004 Měřicí a řídicí technika

Téma:

Měření teplot ve vrtech pro tepelná čerpadla
Temperature Measurement in Borehole Heat Exchangers

Zásady pro vypracování:

1. Rozbor současného stavu měření teplot v geotermálních výměnících pro tepelná čerpadla.
2. Návrh monitorovacího systému pro geotermální výměník ve firmě Greengas.
3. Realizace hardware pro monitorovací systém.
4. Realizace software pro monitorovací systém včetně přenosu dat.
5. Nasazení a testování funkčnosti systému.
6. Zhodnocení výsledků.
7. Rozšířený abstrakt v anglickém jazyce v rozsahu 3 strany sumarizující řešení. Rozšířený abstrakt bude přiložen jako jedna z příloh.

Seznam doporučené odborné literatury:

1. *Studie metodiky měření vlastností horninového masivu v návaznosti na efektivní provoz tepelných čerpadel. Závěrečná zpráva k projektu MPO Efekt 122142 0234, 2011.*
2. Firemní technická dokumentace pro použité komponenty řídicího systému.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Jiří Koziorek, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.



Václav Kubíček

Datum odevzdání diplomové práce: 4. 5. 2012

Poděkování

Děkuji vedoucímu diplomové práce Ing. Jiřímu Koziorkovi, Ph.D. za odborné konzultace a rady, které mi pomohly při vypracování této práce. Také děkuji panu Ing. Zdeňku Rozehnalovi za dobrou spolupráci při tvoření celého projektu. Velké poděkování patří rovněž mým rodičům a blízké rodině za podporu při studiu, Ing. Lumíru Mikolášovi za obětavou korekturu a mé přítelkyni Veronice Mikolášové za pomoc s jazykovou úpravou, překlady anglických textů a psychickou podporu.

Abstrakt

Diplomová práce se zabývá sestavením měřicího a řídicího (monitorovacího) systému pro podzemní zásobník tepla, jeho naprogramováním a uvedením do provozu. Nejprve byl proveden rozbor současného měření teplot v geotermálních výměnících pro tepelná čerpadla a na základě získaných zkušeností byl vytvořen monitorovací systém pro budovaný zásobník. Podzemní zásobník tepla je vytvořen šestnácti energetickými vrty pro vytápění zemské kůry a pěti monitorovacími vrty pro sledování tepelného profilu. Naměřená data jsou odesílána a ukládána na server umístěný v areálu Vysoké školy báňské a budou dále využívána pro další zpracování a vyhodnocování.

Podzemní zásobník je provozován ve dvou režimech – režim nabíjení a režim vybíjení. v režimu nabíjení je zásobník vytápěn přebytečným teplem z plynové kogenerační jednotky a naopak v režimu vybíjení je ochlazován chladicím okruhem.

Úkolem monitorovacího systému je měření teplot v monitorovacích vrtech v požadovaných hloubkách, měření uloženého a odebraného tepla ze zásobníku a jednoduché řízení systému nabíjení a vybíjení.

Pro monitorování a řízení byl vybrán programovatelný logický automat B&R X20 od firmy Bernecker&Rainer. K tomuto automatu jsou připojeny dva druhy teplotních čidel. Pro měření teplot v systému a na vstupech a výstupech vytápěných vrtů jsou použita odporová teplotní čidla PT1000. Pro měření v monitorovacích vrtech jsou použita digitální čidla DALAS komunikující po sběrnici 1-Wire. Měření uložené a odebrané energie se provádí kalorimetrem CALOR 40. Pro komunikaci automatu s kalorimetrem byl vybrán komunikační protokol M-Bus provozovaný na lince RS485. Předmětem řízení tohoto systému je průtok, teplota media vracejícího se do kogenerační jednotky a ovládání čerpadel a ventilátorů.

V současné době je systém v záběhu a testování. Systém je poměrně rozsáhlý a navazuje na jiné systémy, nebylo tedy možné provést jednorázové testování.

Klíčová slova

Programovatelný logický automat, Měření, Regulace

Použité zkratky

PLC	-	programovatelný logický automat
CPU	-	centrální procesorová jednotka

Abstract

The task of this diploma thesis was to assemble a measuring and control (monitoring) system for an underground heat accumulator, create a control program for it and put it into operation. First, an analysis of presently used methods of measuring temperature in geothermal exchangers for heat pumps was carried out and based on data obtained a monitoring system for the currently constructed accumulator was designed.

The heat accumulator is formed by 16 energetic boreholes for heating the Earth's crust and 5 monitoring boreholes for observing the thermal profile. The measured data are sent to and stored at a server located at VŠB. The data will be used for further processing and evaluation.

The underground heat accumulator will operate in two modes – charging mode and discharging mode. During the charging mode the accumulator is heated by the redundant heat from the gas cogeneration unit. On the other hand, during the discharging mode it is cooled by the cooling circuit.

The tasks of the monitoring system are: measurement of temperatures in the monitoring boreholes at the specified depths, measurement of the heat stored at and taken from the accumulator and a simple control system for charging and discharging.

A programmable logic controller B&R X20 was chosen for monitoring and control of the flow rate, the temperature of the medium returning to the cogeneration unit and manipulation with the pumps and ventilators. Two different types of heat sensors are connected to the controller. Resistance temperature sensors PT1000 are used for measuring the temperatures in the system and on the inputs and outputs of the heated boreholes. Digital sensors DALAS communicating over the 1-Wire bus are used for measuring in monitoring boreholes. Calorimeter CALOR 40 was chosen to measure the amount of stored and taken energy. Communication between the controller and the calorimeter is carried out according to the M-Bus protocol on the RS485 line.

The system is being tested under normal operation at the moment. Due to the fact that it is quite extensive and is linked to other systems, it was not possible to carry out a single one-time testing.

Keywords

Programmable Logic Controller, Measurement, Control

Obsah

1	Úvod.....	1
2	Rozbor současného stavu	3
2.1	Související předchozí projekty	3
2.2	Požadavky na vytvářený monitorovací systém	5
2.3	Podzemní zásobník tepla.....	5
2.3.1	Propojení energetických vrtů a umístění teplotních čidel	5
2.3.2	Umístění teplotních čidel v monitorovacích vrtech.....	6
2.4	Popis technologie	6
2.4.1	Režim nabíjení.....	7
2.4.2	Režim vybíjení	7
3	Návrh monitorovacího systému	8
3.1	Vstupy a výstupy procesu	8
3.2	Výběr PLC	8
3.2.1	PLC B&R System X20CP1484.....	9
3.2.2	Rozšiřující moduly použité v této aplikaci.....	10
3.3	Reléové spínače.....	13
3.4	Teplotní čidla PT1000.....	13
3.5	Teplotní čidla DALAS	14
3.6	Kalorimetr CALOR 40.....	15
3.6.1	Druhy podporovaných komunikačních rozhraní	15
3.6.2	Podporované komunikační protokoly	15
3.6.3	Princip komunikace protokolu M-Bus u zařízení CALOR 40 [8]	16
3.7	Čerpadla	17
3.8	Trojcestný ventil s pohonem	17
3.9	Tlakové snímače.....	18
3.10	Projektová dokumentace	18
4	Realizace hardwarové části	18
5	Realizace softwarové části	21
5.1	Automation Basic.....	21
5.1.1	Použitá klíčová slova a funkce v programu.....	21
5.2	Nastavení přepínání letního a zimního času.....	22
5.3	Konfigurace komunikačních rozhraní a vstup/výstupních modulů.....	23
5.4	Přiřazení vstupů a výstupů jednotlivým proměnným.....	24
5.5	Program PLC.....	25
5.5.1	Vyčítání teplot z monitorovacích vrtů.....	25
5.5.2	Čtení hodnot kalorimetru	25
5.5.3	Řízení zpátečky	31
5.5.4	Regulace průtoku.....	34
5.5.5	Rychlá smyčka	37
5.5.6	Řízení systému	37
5.5.7	Vytváření zálohových souborů.....	39

5.5.8	Program obsluhy poruch	44
6	Testování	50
7	Závěr	51

1 Úvod

Diplomová práce se zabývá rozбором současného stavu měření teplot v geotermálních výměnících tepla pro tepelná čerpadla, sestavením měřicího a řídicího systému pro podzemní zásobník tepla, jeho naprogramováním a uvedením do provozu. Systém je určen pro firmu Green Gas DPB, a.s. v Paskově, kde je vytvořen systém 16 energetických a 5 monitorovacích vrtů. Energetické vrty jsou vytápěny tepelným okruhem z plynové kogenerační jednotky, spalující metan odčerpaný s důlních šachet, a následně budou ochlazovány chladicím okruhem. Bude se zjišťovat, kolik tepla se v zásobníku uchová a vyzíská zpět k vytápění. Úkolem vytvářeného systému je řídit nabíjení a vybíjení zásobníku, měřit energii uloženou do zásobníku, nebo z něj odebranou, teploty na vstupech a výstupech topných okruhů jednotlivých energetických vrtů a teplotní profil monitorovacích vrtů a naměřené hodnoty zobrazovat ve vizualizaci a odesílat k archivaci na server, umístěný v areálu Vysoké školy báňské.

Při vypracovávání diplomové práce se vycházelo z poznatků již prováděných monitorování v areálu Vysoké školy báňské, kde jsou vytvořeny hloubkové vrty pro tepelná čerpadla. V jejich okolí jsou vytvořeny polygony pozorovacích vrtů, ve kterých se provádí měření teplotních změn vlivem odčerpávání tepla. Tyto polygony jsou v areálu dva. Na každém polygonu je použit jiný systém pro monitorování.

Na základě těchto zkušeností byl vybrán pro monitorování a řízení řídicí systém B&R X20 od rakouské firmy Bernecker&Rainer. K tomuto systému jsou připojena odporová teplotní čidla PT1000 a digitální čidla DALAS, komunikující po sběrnici 1-Wire. PLC komunikační rozhraní 1-Wire nepodporuje, proto jsou použity převodníky 1-Wire/RS232.

Teploty jsou měřeny na vstupech a výstupech jednotlivých energetických (vytápěných) vrtů o vrtné hloubce 60 metrů, které jsou rozmístěny pravidelně na čtvercové ploše, bylo vhodnější použití odporových teplotních čidel PT1000. V pozorovacích vrtech, rozmístěných okolo zásobníku, jsou teploty měřeny v hloubkách 2, 6 a 15 metrů a v pozorovacím vrtu VM3, umístěném uprostřed zásobníku, o vrtné hloubce 80 metrů jsou teploty měřeny rovněž v hloubkách 2, 6 a 15 metrů, ale dále v hloubkách 30, 40, 50, 60, 70 a 80 metrů. Pro měření v těchto pozorovacích vrtech bylo vhodnější použití digitálních teplotních čidel. Ta jsou namontována na jednom kabelu (tři vodiče) v příslušných vzdálenostech odpovídajících hloubkám měřených teplot. Jednomu pozorovacímu vrtu přísluší jeden kabel a jedno rozhraní 1-Wire.

Součástí monitorování je také měření množství uložené a odebrané energie z podzemního zásobníku tepla. Pro toto měření je použit kalorimetr CALOR 40, který měří energii, výkon, průtok, objem a vstupní, výstupní a rozdílovou teplotu. Pro měření průtoku využívá tento kalorimetr Faradayova zákona elektromagnetické indukce. Aktuální naměřené hodnoty jsou odesílány z kalorimetru do monitorovacího zařízení (PLC) po lince RS485 s využitím komunikačního protokolu M-Bus. Monitorovací zařízení nepodporuje tento komunikační protokol, proto musel být pro monitorovací zařízení vytvořen program obsluhující tuto komunikaci.

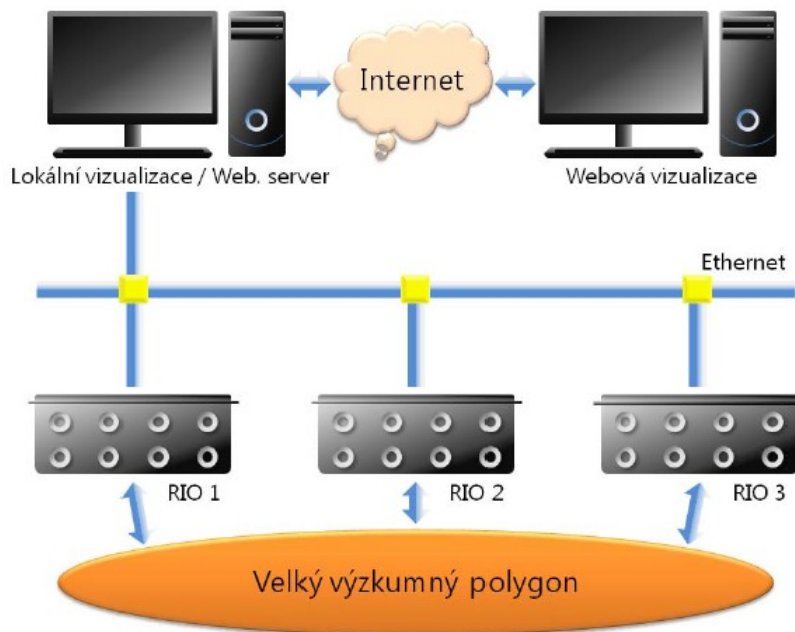
Předmětem řízení je regulace průtoku topného media řízením otáček čerpadla, regulace teploty topného media vracejícího se do kogenerační jednotky trojcestným ventilem a hlídání chybových stavů systému.

2 Rozbor současného stavu

V dnešní době je velmi populárním tématem využití obnovitelných zdrojů energie. Jedním z těchto zdrojů je teplo horninového masivu. Protože je jeho teplotní hladina poměrně nízká, není možné její přímé využití. Toto teplo je však možné převádět na vyšší teplotní hladinu pomocí tepelných čerpadel. Tento systém je již v praxi běžně používán. Nabízí se taky možnost v době přebytku energie, v teplém období, teplo ukládat do zemské kůry a naopak v době nedostatku, v zimním období, jej využívat. Vznikají tedy projekty, kde se ukládá teplo do zemské kůry. Tyto systémy fungují například v Kanadě nebo ve Švédsku. Pro výstavbu podobného projektu se rozhodla firma Green Gas DPB, a.s. sídlící v Paskově, v české republice je to první projekt tohoto typu. Ve spolupráci s Vysokou školou báňskou vytváří podzemní zásobník tepla tvořený hloubkovými vrty, který bude vytápěn (nabíjen) plynovou kogenerační jednotkou, spalující odčerpaný metan z důlních šachet, a následně ochlazován (vybíjen) chladicím okruhem. V oblasti zásobníku jsou vytvořeny monitorovací vrty, ve kterých bude měřen teplotní profil. Teplo ukládané do zásobníku, nebo z něj odebírané, bude měřeno.

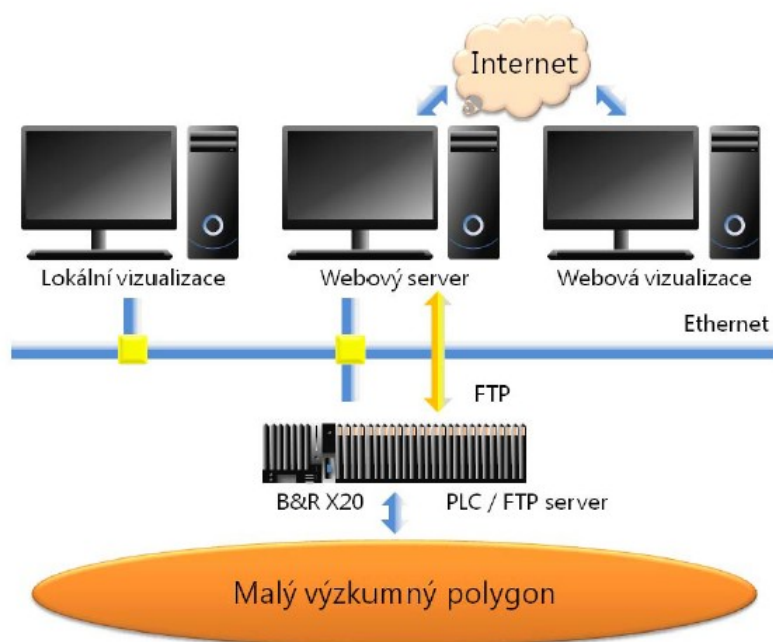
2.1 Související předchozí projekty

V areálu vysoké školy báňské jsou již vytvořeny dva polygony vrtů, velký výzkumný polygon a malý výzkumný polygon, kde se měří teplotní změny horninového masivu vzniklé využíváním jeho tepelné energie pro vytápění pomocí tepelných čerpadel. Je na nich prováděno kontinuální měření teplot v požadovaných hloubkách a tyto teploty jsou ukládány pro další zpracování. Na každém polygonu je použit jiný systém pro měření a sběr dat.



Obr. 1 Blokové schéma měřicího systému na velkém výzkumném polygonu [1]

Na velkém výzkumném polygonu se pomocí systému na bázi PAC od firmy ICPDAS celkově monitoruje v 15 vrtech (10 a 5) na 80 teplotních údajů (60 a 20), měřených pomocí teplotních odporových čidel Pt1000, zapojených čtyřvodičově na vzdálené vstupy a výstupy, a to na svorky měřicích modulů, přičemž na každý z těchto 20 modulů lze připojit až 4 teplotní senzory. Moduly jsou dále sloučeny do tzv. měřicích jednotek obsahujících 8 měřicích modulů. Moduly odesílají naměřená data přes Ethernet na osobní počítač. Na obrázku (Obr. 1) je blokové schéma systému.[1]



Obr. 2 Blokové schéma měřicího systému na malém výzkumném polygonu [1]

Na malém výzkumném polygonu je nainstalován systém na bázi PLC B&R X20 od rakouské firmy Bernecker&Rainer. Doplnující paměť umožňuje ukládání dat do souboru formátu CSV v požadovaných intervalech. Pomocí analogových zásuvných modulů je k PLC přímo připojeno 106 teplotních čidel PT1000 v třívodičovém zapojení. Na tomto systému je dodatečně testováno použití digitálních teplotních čidel DS18B20 (DALAS) komunikujících s monitorovacím zařízením po lince 1-Wire. Použití těchto teplotních čidel je cenově výhodnější a jednodušší než použití analogových čidel PT1000. Na obrázku (Obr. 2) je blokové schéma systému.[1]

Provozem těchto monitorovacích systémů se projevily výhody a nevýhody řešení na jednotlivých výzkumných polygonech. Na velkém výzkumném polygonu dochází při výpadku komunikace mezi měřicí jednotkou a počítačem k ztrátám naměřených dat. Nevýhodou malého polygonu je centrální měřicí zařízení, ke kterému je nutné dovést všechna teplotní čidla. Z důvodu velkého množství teplotních čidel a velké vzdálenosti čidel od měřicí jednotky, musely být skupiny čidel připojeny na mnohožilové kabely. V místě spoje vznikají vlivem

nepříznivých podmínek přechodové odpory a tím dochází k nepřesnému měření. Z ostatních hledisek se monitorovací zařízení na bázi PLC B&R X20 jeví jako vyhovující.

2.2 Požadavky na vytvářený monitorovací systém

Požadované funkce monitorovacího systému:

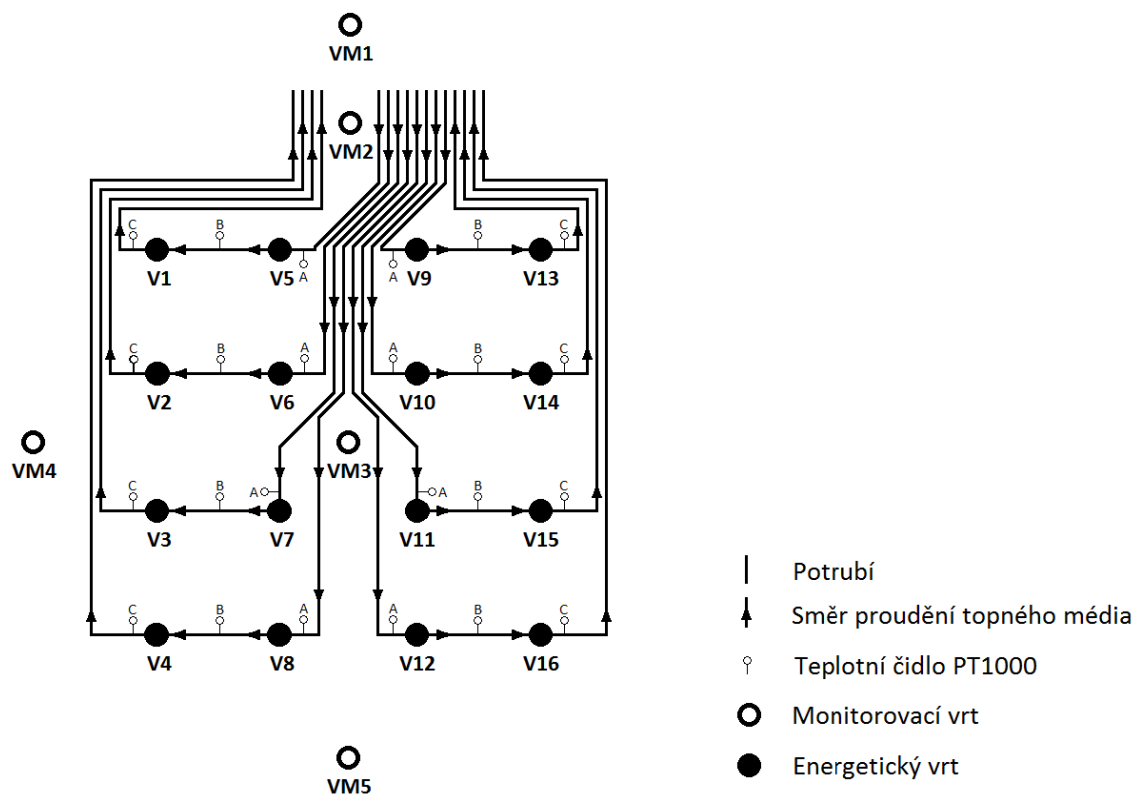
- měření teplot na vstupech a výstupech vytápěných vrtů
- měření teplot v pozorovacích vrtech v požadovaných hloubkách
- měření uložené energie do zásobníku, nebo z něj odebrané
- regulace průtoku topného media
- regulace teploty topného media vracejícího se do kogenerační jednotky
- ovládání ventilátorů
- odesílání naměřených hodnot na server umístěný v areálu Vysoké školy báňské
- hlídání minimálních tlaků v topném a chladícím okruhu
- hlídání poruchových stavů

2.3 Podzemní zásobník tepla

Firma Green Gas vytvořila v otevřené provozní hale systém 21 hloubkových vrtů, z nichž 16 je energetických a 5 je pozorovacích – monitorovacích. Energetické vrty (V) o vrtné hloubce 60 m jsou uspořádány do čtverce (4x4 vrty) a jsou vystrojeny U-trubicemi pro vytápění. Monitorovací vrty (VM) jsou uspořádány okolo zásobníku z energetických vrtů a jeden je umístěn ve středu tohoto zásobníku. Rozmístění vrtů je patrné z obrázku (Obr. 3). **Chyba! Nenalezen zdroj odkazů.**

2.3.1 Propojení energetických vrtů a umístění teplotních čidel

Energetické vrty jsou propojeny vždy dva do série. Prvním vytápěným vrtem ze série je vždy vrt více ve středu zásobníku, aby byl zásobník vytápěn rovnoměrně od středu ke kraji. Teplotní čidla jsou umístěna na vstupech do vrtů, mezi sériově řazenými vrty a na výstupech. v programu jsou teplotní čidla označena čísla vrtů v sérii a písmeny A, B a C. Písmena a označují teplotní čidla na vstupech do vrtů, písmena B čidla mezi vrty v sérii a písmena C označují čidla na výstupech z vrtů.



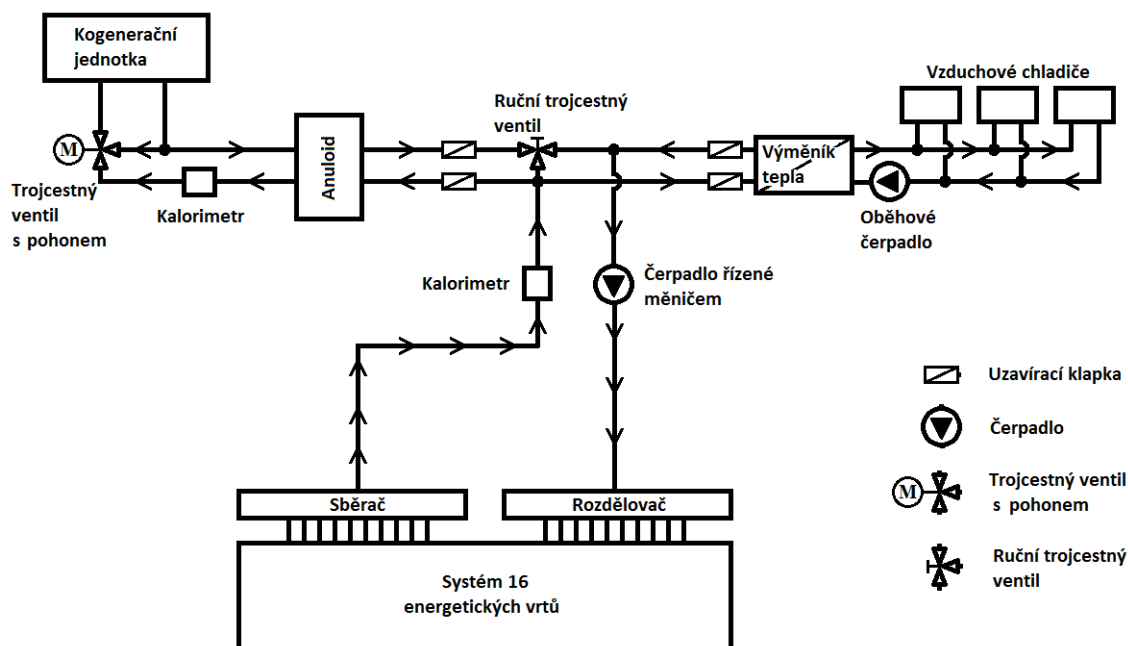
Obr. 3 Rozmístění teplotních čidel mezi vrty.

2.3.2 Umístění teplotních čidel v monitorovacích vrtech

Čtyři monitorovací vrty VM1, VM2, VM4 a VM5 jsou rozmístěny okolo zásobníku. Jejich vrtná hloubka je 15 m a teploty mají být měřeny v hloubkách 2, 6 a 15 metrů pod povrchem. Monitorovací vrt VM3 je umístěn uprostřed zásobníku a je vyvrtán do hloubky 80 metrů. Teplota má být v tomto vrtu měřena v hloubkách 2, 6, 15, 30, 40, 50, 60, 70 a 80 metrů pod povrchem.

2.4 Popis technologie

Systém má dva okruhy. Topný okruh propojený s kogenerační jednotkou a chladicí okruh. Pomocí přepínače v rozvaděčové skříni a uzavíracích klapek u anuloidu a výměníku tepla se přepínají dva režimy - režim nabíjení a režim vybíjení. Principiální schéma technologie je na obrázku (Obr. 4).



Obr. 4 Schéma technologie

2.4.1 Režim nabíjení

K anuloidu je přiváděna topná voda z kogenerační jednotky s teplotou okolo 90°C. Anuloid má funkci vyrovnávače rozdílů tlaků mezi topným systémem kogenerační jednotky a systémem vytápění podzemního zásobníku tepla. Uzavírací klapky u anuloidu jsou otevřeny a u výměníku tepla jsou zavřeny. Pomocí ručního trojcestného ventilu se mísením nastavuje teplota kapaliny vstupující do podzemního zásobníku tepla. Čerpadlem řízeným měničem se nastavuje rychlost proudění média, čímž se reguluje rychlost nabíjení podzemního zásobníku. Za čerpadlem je rozdělovač, rozdělující topné médium do osmi topných okruhů, vždy dvou vrtů v sérii. Za vrty je sběrač, jenž sbírá ochlazenou topnou kapalinu a přivádí ji na kalorimetr, který měří teplo uložené ve vrtech. Dále je kapalina vedena opět na anuloid, pak přes druhý kalorimetr, měřící celkové odebrané teplo pro fakturaci, na trojcestný ventil s pohonem, kterým se reguluje teplota média vracejícího se do kogenerační jednotky.

2.4.2 Režim vybíjení

Při režimu vybíjení jsou uzavírací klapky u anuloidu zavřeny a u výměníku tepla otevřeny. Čerpadlo řízené měničem řídí rychlost proudění topného média, čímž se mění rychlost vybíjení zásobníku. Z čerpadla je přivedena kapalina na rozdělovač rozdělující médium do vrtů. Na výstupu z vrtů sběrač sbírá ohřáté topné médium a přivádí jej na kalorimetr měřící teplo odebrané ze zásobníku. Z kalorimetru jde kapalina do výměníku tepla, kde předá teplo chladicímu okruhu. V chladicím okruhu je oběhové čerpadlo, které vytváří proudění chladicí nemrznoucí směsi mezi výměníkem tepla a vzduchovými chladiči. V okruhu jsou instalovány tři vzduchové chladiče s ventilátory.

3 Návrh monitorovacího systému

Při návrhu monitorovacího systému se vycházelo ze zkušeností z projektů vytvořených v areálu Vysoké školy báňské. Požadavky na měření teplot byly srovnatelné s požadavky na monitorovací systém instalovaný ve škole. Rozšířením systému je komunikace s kalorimetrem, kontrola tlaků v okruzích a ovládání čerpadel a ventilátorů.

3.1 Vstupy a výstupy procesu

Vstupy:		typ	napětí
○ teplotní čidla PT1000	27x	odporový	
○ teplotní čidla DALAS	5 skupin čidel	digitální kom. 1-Wire	
○ kalorimetr CALOR 40		RS485	
○ pokles tlaku pod stanovenou mez – hl. okruh		dvoustavový spínací	
○ pokles tlaku pod stanovenou mez – chl. okruh		dvoustavový spínací	
○ tlačítko kvitování poruchy		dvoustavový spínací	
○ přepínač nabíjení/vybíjení – nabíjení		dvoustavový spínací	
○ přepínač nabíjení/vybíjení – vybíjení		dvoustavový spínací	
○ dobíjení záložní baterie		dvoustavový spínací	
Výstupy:			
○ spuštění čerpadla hlavního okruhu		dvoustavový	230V
○ aktivování frekvenčního měniče		dvoustavový	230V
○ regulace čerpadla v hlavním okruhu		spojitý	0 – 10V
○ spuštění oběhového čerpadla chladicího okruhu		dvoustavový	230V
○ otevírání trojcestného ventilu - vstup Y1		dvoustavový	230V
○ zavírání trojcestného ventilu - vstup Y2		dvoustavový	230V
○ spuštění ventilátorů 1		dvoustavový	230V
○ spuštění ventilátorů 2		dvoustavový	230V
○ spuštění ventilátorů 3		dvoustavový	230V
○ kontrolka poruchy		dvoustavový	230V

3.2 Výběr PLC

Z důvodu programování a pozdější údržby bylo vybíráno z automatů, které mají podporu na škole, což jsou PLC od firem SIEMENS, Bernecker&Rainer a Rockwell. Dalším kritériem byla podpora vybavení pro monitoring. Standardní vstupy a výstupy pro jednoduché řízení podporují všechna zařízení, avšak vstupy pro měření teplot se lišily. PLC X20 System má možnost připojení zásuvných modulů pro čidla PT1000, kde je výstupem přímo hodnota teploty v °C. U ostatních zařízení je potřeba čidlo připojit na analogový vstup a vypočítávat teplotu v programu PLC. PLC X20 System splňuje i řadu dalších požadavků jako ethernetové připojení, rozhraní RS232 a jiné. Pro řízení bylo tedy vybráno zařízení PLC X20 System.

3.2.1 PLC B&R System X20CP1484

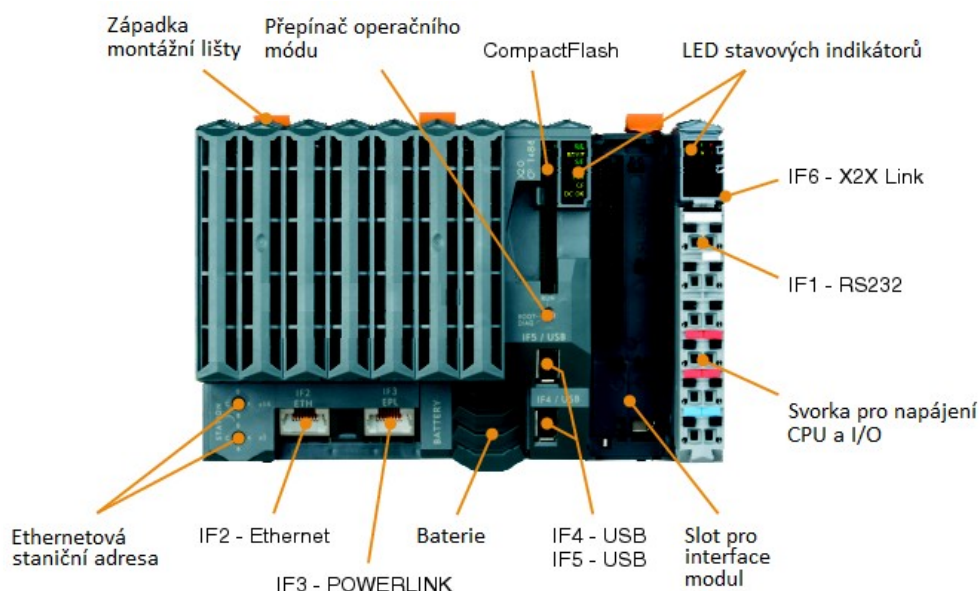
Nová řada procesorů X20 System splňuje řadu požadavků. Tyto procesory mohou být implementovány kdekoli, od standardních aplikací až po ty nejnáročnější s vysokými výkonnostními požadavky. Může zvládnout efektivně i 200 μ s cykly.[2]

RS232, Ethernet a USB jsou v B&R standardní výbavou. Možnost práce v síti a připojení USB zařízení je tedy možné bez dodatečných nákladů. Navíc má každý procesor POWERLINK připojení pro komunikaci v reálném čase. Ačkoli standardní procesorová jednotka splňuje většinu požadavků, jsou k dispozici až tři sloty pro přídavné interface moduly. [2]

Protože procesory X20 jsou navrženy pro instalaci na montážní lištu v rozvaděči, může být přímo připojeno až 250 vstup/výstupních X20 modulů (3000 kanálů). Avšak nejvyšší výkon poskytuje i při použití výhod vzdálených modulů. [2]

Na základě technologie nejnovějšího procesoru Intel Celeron mohou procesory X20 využívat 200 μ s cykly. Velké množství paměti SDRAM poskytne uživateli volnost při tvoření aplikace. Tato paměť je doplněna vyrovnávací permanentní pamětí SRAM, napájenou baterií, pro specifická tasková data a remanentní proměnné. V případě výpadku napájení jsou proměnné, které byly deklarovány jako remanentní, automaticky zkopírovány z rychlé paměti RAM na zabezpečenou SRAM. Data jsou uchována až do restartu procesoru. Procesorová jednotka je také vybavena slotem pro CompactFlash k ukládání programu a aplikačních dat. [2]

Pro všechny platformy je potřebný pouze programovací nástroj B&R Automation Studio. Pro vytváření aplikačního softwaru mohou být použity všechny základní jazyky normy IEC61131-3 a jazyk C. Integrovaná vizualizace, funkce NC a soft CNC a Web server technologie doplňují řadu uživatelských funkcí. [2]



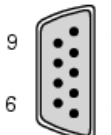

Obr. 5 Základní modul PLC X20CP1484 [2]

Parametry použitého PLC X20CP1484:

- Procesor – Intel Celeron 266
- operační paměť - 32 MB SDRAM a 1 MB SRAM.
- nejkratší doba cyklu je 800 μ s

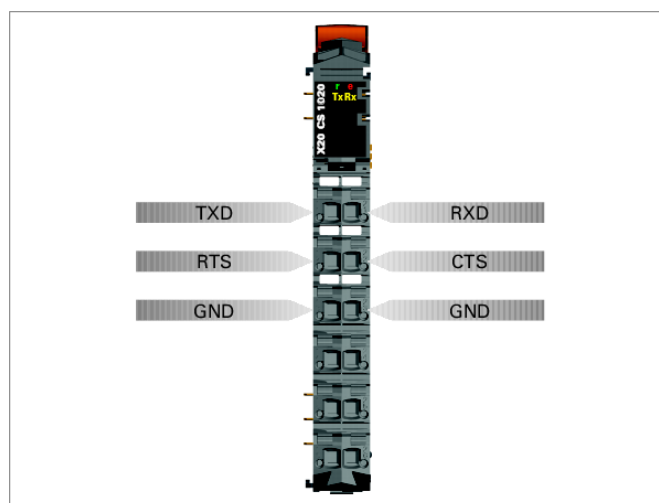
3.2.2 Rozšiřující moduly použité v této aplikaci

Modul IF1030 - do slotu pro interface modul je připojen modul pro rozhraní RS485/RS422. Toto rozhraní je použito pro komunikaci s kalorimetrem CALOR 40. V horní části modulu jsou tři stavové LED. Pokud je modul aktivní a je vše v pořádku svítí zelená LED STATUS. v případě, že CPU startuje, svítí LED STATUS červeně. LED RxD, nebo TxD svítí žlutě, pokud modul vysílá, nebo přijímá přes rozhraní RS485/RS422. Na obrázku (Obr. 6) jsou popsány jednotlivé kontakty připojovacího konektoru.

Interface	Pin assignments			Figure
Application interface RS485/RS422  9-pin DSUB socket	Pin	RS485	RS422	
	1	Reserved	Reserved	
	2	Reserved	TxD (1)	
	3	DATA	RxD	
	4	Reserved	Reserved	
	5	GND	GND	
	6	+5 V / 50 mA	+5 V / 50 mA	
	7	Reserved	TxD\ (1)	
	8	DATA\	RxD\	
	9	Reserved	Reserved	

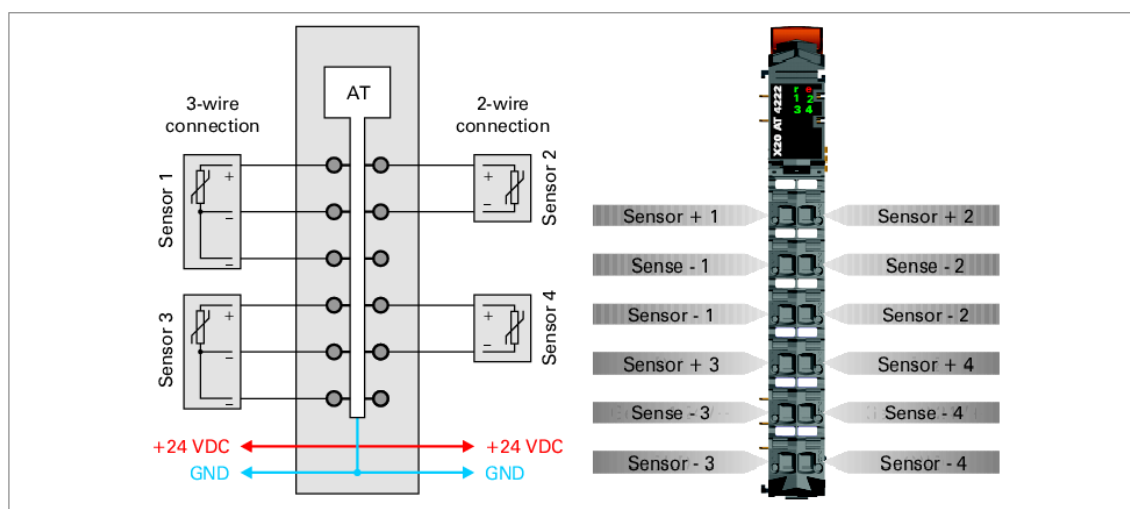
Obr. 6 Zapojení modulu IF1030 [2]

Moduly CS1020 - na sběrnici X2Xlink jsou připojeny čtyři zásuvné moduly s rozhraním RS-232 komunikující s převodníky ADA-101W. V horní části modulu jsou čtyři stavové LED. Zelené „r“ svítí, pokud je vše v pořádku. Červené „e“ svítí v případě poruchy. Žluté Tx a Rx svítí v případě vysílání nebo přijímání zprávy přes rozhraní RS232. Na obrázku (Obr. 7) jsou označeny kontakty.



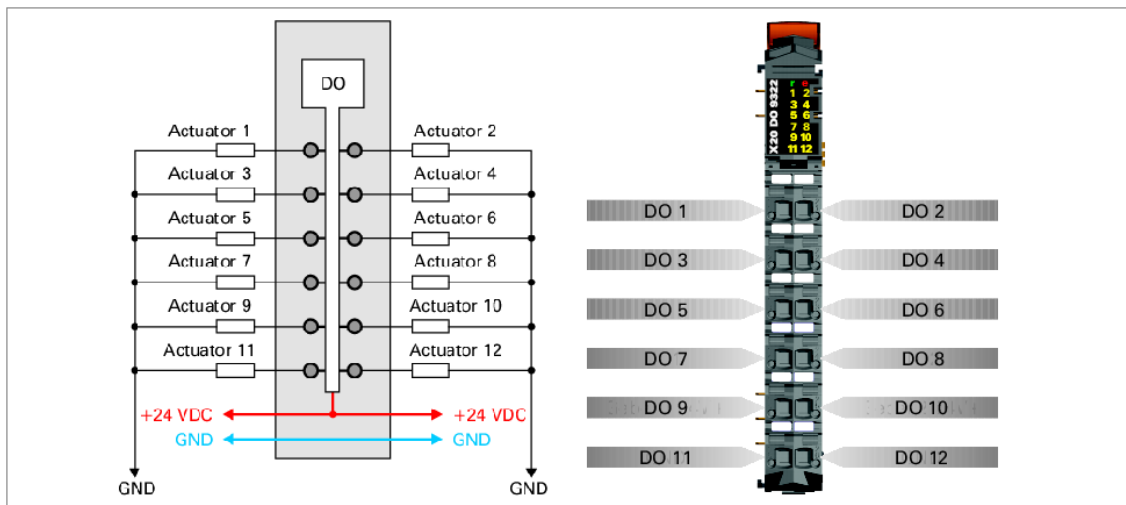
Obr. 7 Zapojení modulu CS1020 [2]

Moduly AT4222 – na sběrnici X2Xlink je připojeno osm zásuvných modulů pro měření teploty. Každý modul má čtyři kanály pro připojení čidel PT1000/PT100 ve dvou, nebo třívodičovém zapojení. Každý kanál lze nakonfigurovat na příslušné čidlo a typ jeho zapojení. Na obrázku (Obr. 8) je příklad dvouvodičového (senzory 2 a 4) a třívodičového (senzory 1 a 3) zapojení.



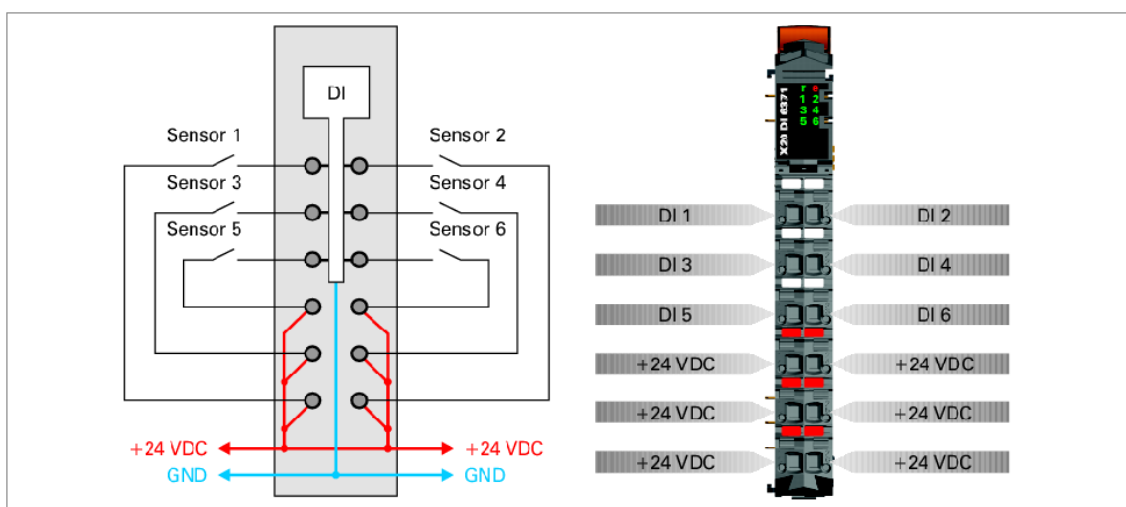
Obr. 8 Zapojení modulu AT4222 [2]

Modul DO9322 – na sběrnici X2Xlink je připojen jeden zásuvný modul s dvanácti digitálními výstupy. Modul obsahuje pouze kladnou svorku výstupního kanálu. Obvod se uzavírá přes akční člen proti společné zemi napájení PLC. Výstupy jsou určené pro výkonové akční členy s proudovými odběry až 0,5 A. Na obrázku (Obr. 9) je příklad zapojení výstupů.



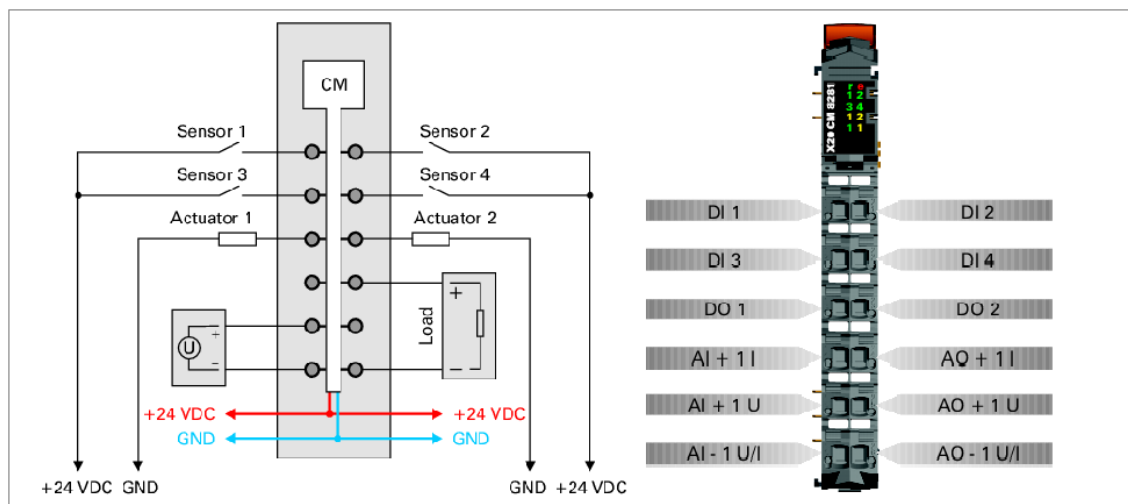
Obr. 9 Zapojení modulu DO9322 [2]

Modul DI6371 – na sběrnici X2Xlink jsou připojeny dva moduly s digitálními vstupy. Každý modul má šest vstupů. Vstup reaguje na 24V napěťový potenciál vztažený k společné zemi napájení PLC. Modul také obsahuje šest svorek s kladným 24V napětím pro realizaci vstupů spínacími kontakty. Na obrázku (Obr. 10) je příklad zapojení vstupů se spínacími kontakty.



Obr. 10 Zapojení modulu DI6371 [2]

Modul CM8281 – na sběrnici X2Xlink je připojen zásuvný modul se čtyřmi digitálními vstupy, dvěma digitálními výstupy, jedním analogovým vstupem a jedním analogovým výstupem, které lze zapojit jako proudovou, nebo napěťovou smyčku. U tohoto modulu je použit pouze analogový výstup. Využívá se pro regulaci frekvenčního měniče. Výstup je zapojen pro napěťovou smyčku -10/10V. Na obrázku (Obr. 11) je příklad zapojení vstupů a výstupů.



Obr. 11 Zapojení modulu CM8281 [2]

3.3 Reléové spínače

Řídicí automat pracuje s napětovým rozsahem 24V, ale stykače pro čerpadla a ventilátory potřebují ovládací napětí 230V. Proto jsou použita spínací relé pro spínání síťového napětí. Ovládací cívka tohoto relé je ovládána napětím 24V. Na obrázku (Obr. 12) je schéma připojovacích kontaktů.



Obr. 12 Spínací relé

3.4 Teplotní čidla PT1000

Čidla PT1000 jsou platinové odporové snímače teploty. Teplota snímače se definuje na základě jeho elektrického odporu. Vzorce pro výpočet elektrického odporu snímače jsou:

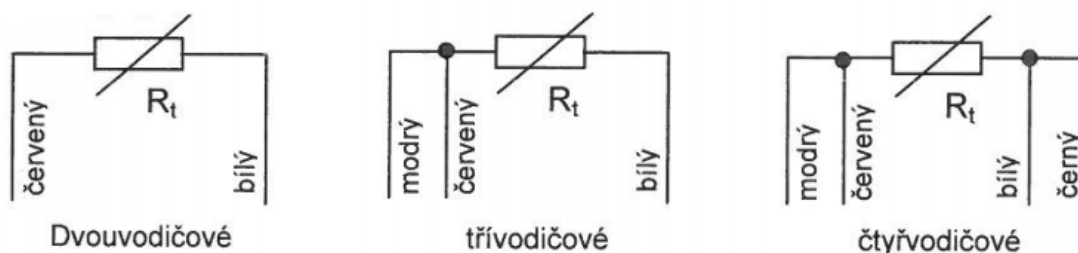
$$\begin{aligned} \text{pro teploty } -50\text{ }^{\circ}\text{C až } 0\text{ }^{\circ}\text{C} \quad R &= R_0 * (1 + \alpha t + \beta t^2 + \gamma[t - 100]t^3), \\ \text{pro teploty } 0\text{ }^{\circ}\text{C až } 400\text{ }^{\circ}\text{C} \quad R &= R_0 * (1 + \alpha t + \beta t^2), \end{aligned}$$

kde

$$\begin{aligned} R_0 &= 1000\ \Omega, \\ \alpha &= 3,9083 * 10^{-3}\text{ }^{\circ}\text{C}^{-1}, \\ \beta &= -5,775 * 10^{-7}\text{ }^{\circ}\text{C}^{-2}, \\ \gamma &= -4,183 * 10^{-12}\text{ }^{\circ}\text{C}^{-4} \end{aligned}$$

[3]

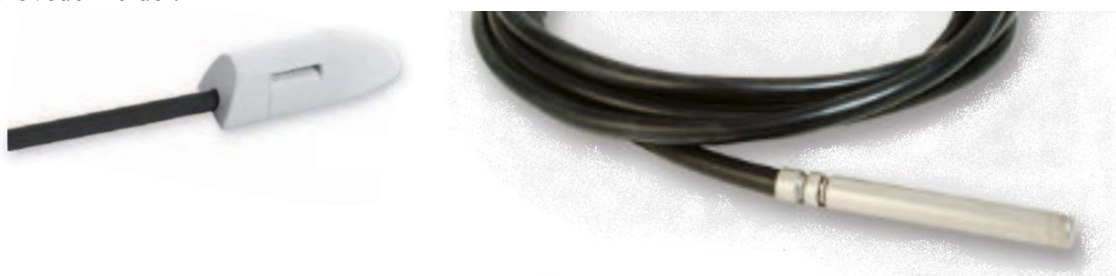
Ze vzorce plyne, že s rostoucí teplotou se zvyšuje elektrický odpor snímače. Stejnou vlastnost jako teplotní snímač mají i přívodní vodiče, což by do měření zavádělo značnou chybu. Proto se pro měření používá dvou, tří, nebo čtyřvodičové zapojení. Dvouvodičové zapojení se používá na velmi krátkých vzdálenostech přívodních vodičů, kde se odpor vodičů neprojeví, nebo není zapotřebí velké přesnosti měření. Na větší vzdálenosti se používá tří, nebo čtyřvodičové zapojení, kde dva vodiče jsou pro proudovou smyčku a jeden, nebo dva vodiče jsou pro napěťovou smyčku. Schémata zapojení výše uvedených konfigurací jsou na obrázku (Obr. 13).



Obr. 13 Zapojení čidel PT1000

V této aplikaci jsou čidla připojena na PLC X20 k zásuvným modulům AT4222, které podporují dvou nebo třívodičová zapojení. Vzhledem k délce přívodního vedení teplotních čidel bylo vybráno třívodičové zapojení. Zapojení vodičů k modulu AT4222 je zobrazeno na obrázku (Obr. 8).

Tato teplotní čidla jsou použita pro měření teplot na vstupech a výstupech energetických vrtů, na zpátečce do kogenerační jednotky a na vstupu a výstupu chladiců v chladicím okruhu. Jsou použita dvě provedení. Příložné provedení umožňuje připevnit čidla pomocí kovových pásek přímo na potrubí, což je použito na vstupech a výstupech vrtů, a jímkové provedení umožňující montáž do jímky použité na zpátečce do kogenerační jednotky a vstupu a výstupu chladicího okruhu. Na obrázku (Obr. 14) je vlevo příložné a vpravo jímkové provedení čidel.



Obr. 14 Čidla PT1000

3.5 Teplotní čidla DALAS

Jedná se o digitální teploměry s rozhraním 1-Wire. Umožňují měřit teplotu v rozsahu od -55°C až do $+125^{\circ}\text{C}$ s přesností $\pm 0,5^{\circ}\text{C}$ garantovanou v rozsahu od -10°C až do $+85^{\circ}\text{C}$. Samotný rozsah dat teplotních čidel je možno měnit od 9 do 12 bitů. Každé čidlo má své unikátní 64 bitové sériové číslo, uchovávané v ROM paměti, které slouží k adresaci konkrétního senzoru. Toto číslo usnadňuje rozpoznání čidel a následnou komunikaci. 64 bitové sériové číslo

společně s unikátním protokolem identifikace připojených zařízení ke sběrnici umožňuje provozovat několik senzorů na jednom datovém vodiči. Samotná čidla obsahují tři výstupy, kde dva slouží pro napájení (GND a VDD) a třetí (DQ) slouží pro komunikaci po 1-Wire sběrnici. Napájecí napětí se pohybuje v rozmezí od 3,3V až po 5,5V. Pro případ dvou vodičového (pouze GND a DQ) propojení se sítí 1-Wire obsahují kondenzátory (C_{pp}), které slouží jako nábojové pumpy a dodávají obvodu společně s datovým vodičem dostatečné napájecí napětí pro plnohodnotnou činnost.[4]

Tato teplotní čidla jsou sestavena do „série“ na jednom kabelu pro každý monitorovací vrt v příslušných vzdálenostech. Tyto kabely jsou spuštěny do vrtů, kde měří teploty v požadovaných hloubkách.

PLC nepodporuje rozhraní 1-Wire, proto jsou použity převodníky ADA-101W, které převádějí komunikaci 1-Wire na RS232. Pro každý monitorovací vrt (jeden kabel) je jeden převodník připojený k PLC pomocí rozhraní RS232. Tuto problematiku řešil Ing. Tomáš Vavrila jako svou diplomovou práci, proto byla do projektu implementována tímto řešitelem.

3.6 Kalorimetr CALOR 40

Měřič tepla CALOR 40 je založen na principu měření známým Faradayovým zákonem elektromagnetické indukce, podle kterého se při proudění elektricky vodivé kapaliny přes magnetické pole průtokoměru indukuje elektrické napětí. To je snímáno dvěma elektrodami, které mají přímý kontakt s měřeným médiem, a vyhodnocováno v elektronické jednotce. Kalorimetr měří energii, objem, výkon, průtok a vstupní, výstupní a rozdílovou teplotu. Hodnoty energie a objemu jsou ukládány do denního a měsíčního archívu.[5]

Měřiče CALOR 40 podporují několik komunikačních protokolů a fyzických rozhraní.

3.6.1 Druhy podporovaných komunikačních rozhraní

- M-Bus
- RS485
- RS422
- IrDa
- GSM

[5]

V této aplikaci je použito rozhraní RS485, které se (stejně jako RS422) vyznačuje dvou vodičovým propojením jednotek. Tyto vodiče se označují písmeny A a B, někdy se používá označení „-“ a „+“. V klidovém stavu by na vodiči A (neboli „-“) mělo být menší napětí než na vodiči B (neboli „+“). Maximální délka sběrnice je až 1200 m, maximální počet uzlů (tj. zařízení vysílajících a přijímajících po lince data) je 32. Při použití opakovačů může být počet uzlů vyšší. Maximální přenosová rychlost je nepřímo úměrná délce vedení. Přenosová rychlost u krátkých spojů (do 10 m) může být až 10 Mb/s. [6]

3.6.2 Podporované komunikační protokoly

- MBus

- Amset
- CALOR30 (tento protokol je zde jen z důvodu kompatibility – další podpora ukončena) [5]

V této aplikaci je použit průmyslový komunikační protokol M-Bus (také Meter Bus) určený především pro dálkový odečet hodnot z měřičů spotřeby, kde příliš nezáleží na rychlosti komunikace jako spíše na odolnosti proti rušení. Po sběrnici probíhá asynchronní sériová 8bitová komunikace, kdy si jednotlivé strany posílají ucelené rámce. Komunikace probíhá způsobem Master-Slave.[7]

3.6.3 Princip komunikace protokolu M-Bus u zařízení CALOR 40 [8]

Na sběrnici je jedno zařízení Master a až 250 zařízení Slave. Každé Slave zařízení má adresu, pomocí které zařízení Master zasílá zařízení Slave požadavek k zaslání zprávy s naměřenými údaji. Komunikace je sériová a probíhá v bajtových rámcích.

Master vyšle ucelený rámec bajtů, který určuje které zařízení má odpovídat a jakou akci má provést. Rámec pro vyžádání aktuálních hodnot má tuto posloupnost bajtů. V závorkách je prezentována hodnota bajtu v hexadecimálním kódu. První bajt zprávu zahajuje (10h), druhý bajt určuje, že se má zaslat aktuální naměřené hodnoty (5Bh nebo 7Bh), následuje adresa dotazovaného zařízení, pak bajt s kontrolním součtem a stop bajt (16h).

Tuto zprávu poslouchají všechna zařízení a zařízení s odpovídající adresou odpoví. Rámec o sedmdesáti devíti bajtech opět začíná startovacím bajtem (68h) a dvěma bajty s délkou rámce (49h, nebo 4Ah). Pak následuje opět startovací bajt (68h), pak následuje „prázdný“ bajt (08h), adresa zařízení, kontrolní informace (72h) informující o rámci s variabilními daty, další čtyři bajty obsahují identifikační číslo zařízení, dva bajty s informací o výrobci, bajt s informací o jaké zařízení se jedná a patnáctý bajt nese informaci o druhu měřidla. Následující je bajt načítající počet přístupů k zařízení, pak bajt s chybovými hláškami a dva bajty se signaturou. Dvacátým bajtem začíná rámec s naměřenými daty, kde první bajt informuje o počtu bajtů dané proměnné, druhý bajt nese informaci o násobku jednotek a pak následuje daný počet bajtů proměnné. První proměnnou je energie, následuje objem, výkon a průtok. Všechny čtyři proměnné jsou přenášeny jako 32bitová celá čísla. Dále jsou vstupní, výstupní a rozdílová teplota, které jsou přenášeny jako 32bitové reálné hodnoty. Následující čtyři bajty přenášejí časový bod čtení hodnot, další dva bajty nesou informaci o celkové době měření. Posledních osm bajtů nese informaci o verzi softwaru, specifických datech měřidla, alarmech, kontrolním součtu a rámec končí stop bajtem.

V této aplikaci je kalorimetr využíván pro měření dodaného, ale i odebraného tepla. Avšak použitý kalorimetr byl navržen pro měření komerčních dodávek tepla, proto neuměl pracovat se záporným rozdílem teplot, a tedy nepodporoval funkci odečtu energie. Proto byla u výrobce kalorimetru vyžádána úprava softwaru měřiče, aby byla tato funkce podporována. Do výčtu proměnných byla tedy přidána nová proměnná energie, která naopak pracuje jen se zápornou hodnotou rozdílu teplot. V bajtovém rámci je proměnná umístěna přímo za původní proměnnou, čímž se zbytek rámce posunul o šest bajtů a jeho délka se prodloužila na osmdesát pět bajtů.

Protokol podporuje také nastavování některých proměnných, například času a vyčítání denních nebo měsíčních archivů.

Nastavení času se provádí zasláním rámce o šestnácti bajtech, kde jsou informace o požadované akci, adrese zařízení, datu a času. Nastavované zařízení v případě úspěšného nastavení potvrdí provedení zasláním potvrzovacího bajtu.

Vyčtení archívu se provádí rovněž zasláním bajtového rámce, ale o délce dvanácti bajtů. Opět je zde informace o délce rámce, požadovaná akce, adresa zařízení, adresa místa čtení v paměti zařízení a kontrolní součet. Na tuto zprávu kalorimetr odpovídá potvrzovacím bajtem (E5h) a pak čeká čtyři vteřiny na požadavek k odeslání informací z archívu. Požadavek má stejný formát jako požadavek na vyčtení aktuálních měřených hodnot. Pokud požadavek nepříjde do čtyř vteřin, nastaví se kalorimetr opět do standardního režimu pro vyčítání aktuálních hodnot. [8]

Pomocí rozhraní RS485 se každou vteřinu vyčítají všechny aktuální měřené hodnoty, které se ukládají. Hodnota aktuálního průtoku je využívána pro regulaci průtoku.

3.7 Čerpadla

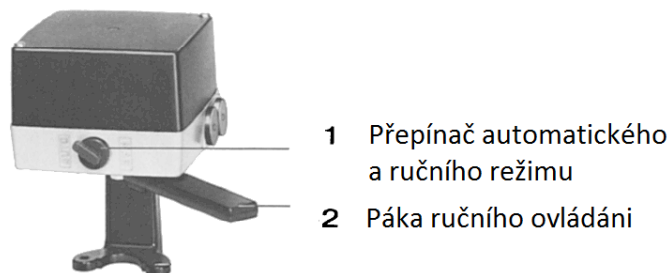
V aplikaci jsou dvě čerpadla. V hlavním okruhu je čerpadlo řízené frekvenčním měničem a v chladicím okruhu je oběhové čerpadlo spouštěné digitálním výstupem z PLC.

Čerpadlo v hlavním okruhu s třífázovým asynchronním motorem je možné spustit v automatickém, nebo ručním režimu. v ručním režimu je motor napájen přímo ze sítě. v režimu automatickém je motor řízen frekvenčním měničem od firmy Siemens. Pro ovládání měniče je potřeba dva digitální výstupy z PLC a jeden analogový. Prvním výstupem je sepnuto napájení měniče. Poté je potřeba počkat 20 vteřin než se inicializuje software měniče. Následně je druhým výstupem z PLC spuštěn start motoru. Analogovým výstupem z PLC je pomocí napěťové smyčky 0-10V řízena frekvence napětí na výstupu měniče, čímž se mění otáčky čerpadla.

Oběhové čerpadlo v chladicím okruhu je spouštěno stykačem ovládaným digitálním výstupem z PLC.

3.8 Trojcestný ventil s pohonem

Trojcestným ventilem se reguluje teplota topného média vracejícího se do kogenerační jednotky mísením ochlazeného média s médiem z přívodu tak, aby teplota neklesla pod stanovenou mez. Ventil je přestavován elektrickým pohonem SQK33. Pohon má dva vstupy Y_1 a Y_2 . Pokud připojíme signál na svorku Y_1 , ventil otevírá zpátečku a uzavírá propojení od přívodu. Pokud připojíme signál na svorku Y_2 , ventil uzavírá zpátečku a otevírá propojení od přívodu. Pohon nemá žádnou zpětnou vazbu pro zjištění aktuální polohy ventilu. Přestavení ventilu o 90°, tedy z polohy 1 do polohy 2, trvá 125 sekund. Servopohon je na obrázku (Obr. 15).



Obr. 15 Servopohon trojcestného ventilu

3.9 Tlakové snímače

Snímače jsou dvoustavové. Jeden snímač je instalován v hlavním okruhu a druhý je v okruhu chladicím, kde se kontroluje, zda neklesl tlak topného nebo chladicího média. Snímače mohou sepnout výstupy buď při překročení nastavené hodnoty tlaku, nebo naopak při poklesu pod nastavenou hodnotu. U snímačů lze nastavovat hodnotu tlaku, na kterou snímač reaguje, a hysterezi této hodnoty.

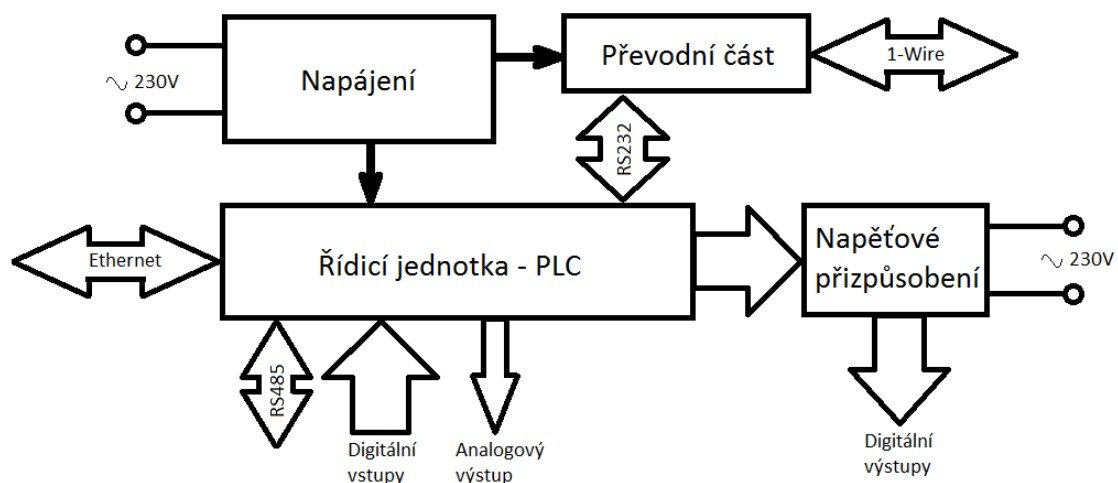
V této aplikaci jsou snímače použity pro detekci nízkého tlaku, spínače se tedy sepnou při poklesu tlaku pod nastavenou mez.

3.10 Projektová dokumentace

K celému projektu byla projektanty vytvořena projektová dokumentace, která měla dvě verze. V první verzi byla pro měření teplot v monitorovacích vrtech navržena odporová teplotní čidla PT1000. Při dalším postupu výstavby byl tento návrh přehodnocen a změněn na digitální teplotní čidla DALAS, komunikující s měřicím systémem po lince 1-Wire.

4 Realizace hardwarové části

Řídicí systém byl navržen tak, že se vytvoří řídicí deska, která se namontuje do rozvaděče a napojí na silnoproudou část systému. Dle části projektové dokumentace Monitorovací systém byly nakoupeny komponenty pro řídicí desku. Na montážní desku o rozměrech 750x550 mm jsou namontovány tři DIN lišty, na které jsou namontovány všechny komponenty. Drátové propojení celého řídicího systému je provedeno v perforovaných žlabech o rozměrech 40x40 mm. Řídicí desku lze rozdělit do čtyř základních bloků (částí). Na obrázku (Obr. 16) je blokové schéma řídicí desky a na obrázku (Obr. 17) je její fotografie.



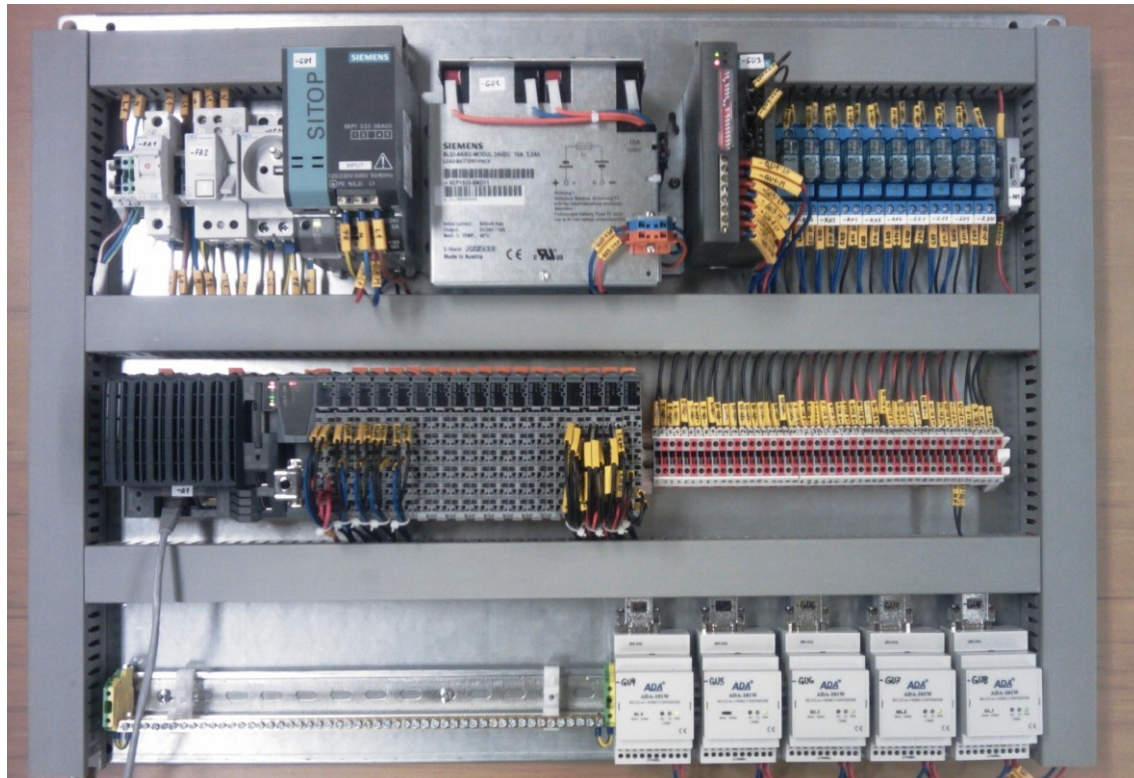
Obr. 16 Blokové schéma řídicí desky

Napájecí část obsahuje šestiampérový jistič, proudový chránič, zásuvku, napájecí zdroj 24V a UPS jednotku s baterií.

Řídicí část je tvořena automatem B&R X20 se zásuvnými moduly

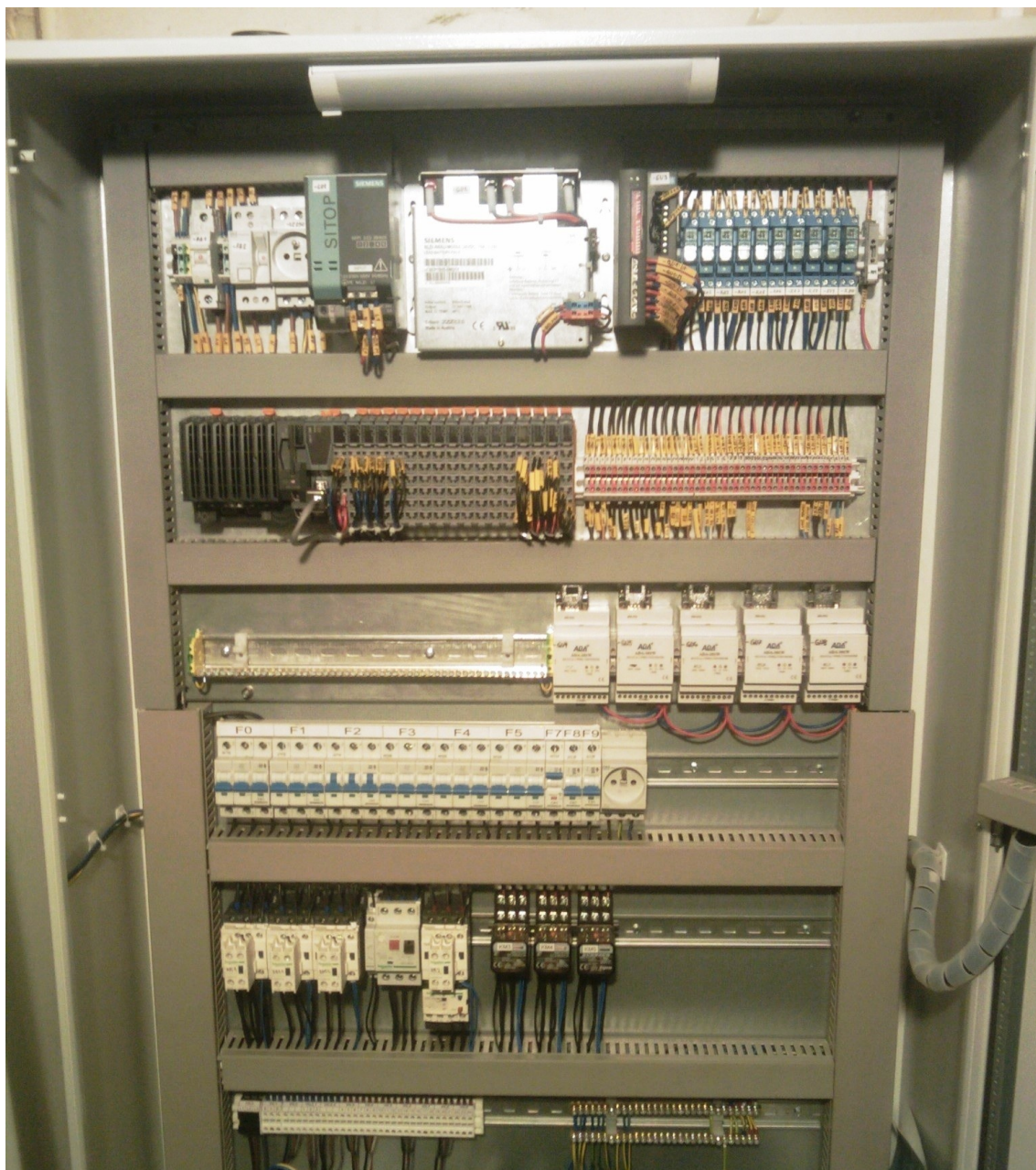
Převodní část obsahuje převodníky ADA-101W převádějící komunikaci RS232 na 1-Wire.

Napěťové přizpůsobení je vytvořeno relátko, která jsou ovládána výstupy z automatu a spínají síťové napětí pro ovládání stykačů a pohonu.



Obr. 17 Fotografie sestavené řídicí desky

Sestavena řídicí deska byla namontována do rozvaděčové skříně celého systému. Bylo provedeno napojení silnoproudé části systému, připojení řídicí desky k napájecí síti a k Ethernetu. Dále bylo provedeno natažení teplotních čidel a jejich napojení. Na obrázku (Obr. 18) je fotografie namontované a zapojené řídicí desky.



Obr. 18 Fotografie namontované řídicí desky

5 Realizace softwarové části

Program PLC je naprogramován v programovacím jazyku Automation Basic a je rozdělen do více částí, tedy podprogramů.

5.1 Automation Basic

Programovací jazyk Automation Basic je vytvořen vývojáři firmy Bernecker&Rainer. Odvíjí se od programovacího jazyka Basic. Při programování automatizačních aplikací se využívá jednoduchých funkcí, proto naučit se velmi rychle tento jazyk není pro programátora příliš obtížné.

Od programovacího jazyka C se Automation Basic v základu liší klíčovými slovy, nevyužívá složených závorek, ty jsou nahrazeny klíčovými slovy pro konce funkcí, a konce řádků se neukončují středníky.

Jazyk má vytvořenou podporu základních funkcí pro práci s pamětí, paměťovými oblastmi, řetězci a proměnnými. Dále jsou vypsány použité funkce a klíčová slova.

5.1.1 Použitá klíčová slova a funkce v programu

INC() – inkrementuje proměnnou v závorce o jedničku.

ABS() – vrátí absolutní hodnotu z čísla v závorce.

ADR() – vrátí adresu proměnné v paměti

SHR(,) – provede bitový posun doprava o požadovaný počet bitů

IF – podmínka. Obdoba podmínky if v programovacím jazyku C. Je možné používat i příkazy ELSE a ELSE IF.

```
Příklad:      IF <podmínka> THEN
               <program>
               ENDIF
```

CASE – větvení programu. Na základě hodnoty vstupní proměnné se vybírá která část programu (akce) se provede.

```
Příklad:      CASE <vstupní proměnná> OF
               ACTION <parametr_1> :
                   <program_1>
               ENDACTION
               ACTION <parametr_2> :
                   <program_2>
               ENDACTION
               ENDCASE
```

LOOP – cyklus. Obdoba cyklu for v programovacím jazyku C. Nedá se však nastavovat krok proměnné určující počet cyklů.

```
Příklad:      LOOP L = 1 TO 38 DO
               <program>
               ENDLOOP
```

FUB – volání funkce se vstupními a výstupními parametry ve struktuře.

Příklad: `FRM_gbuf_01 FUB FRM_gbuf ()`

`atof()` – převádí řetězec na reálné číslo.

`atoi()` – převádí řetězec na celé číslo

`ftoa()` – převádí reálné číslo na řetězec

`itoa()` – převádí celé číslo na řetězec

Příklad: `itoa(int_value, ADR(string))`

`memcpy()` – kopíruje jednu paměťovou oblast do druhé

Příklad: `memcpy(ADR(Mem_Dest), ADR(Mem_Src), length)`

`strcat()` – vkládá řetězec za jiný řetězec

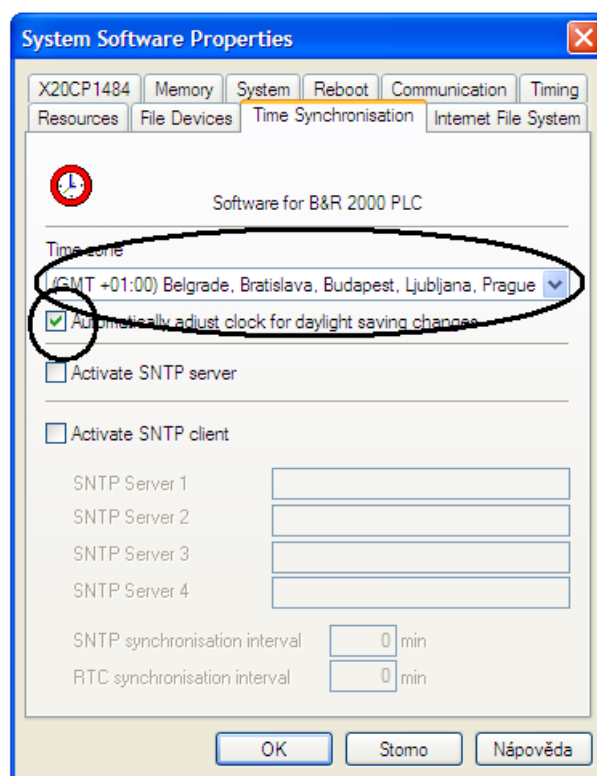
Příklad: `strcat(ADR(string1), ADR(string2))`

`strcpy()` – kopíruje text do řetězce

Příklad: `strcpy(ADR(string1), "Example")`

5.2 Nastavení přepínání letního a zimního času

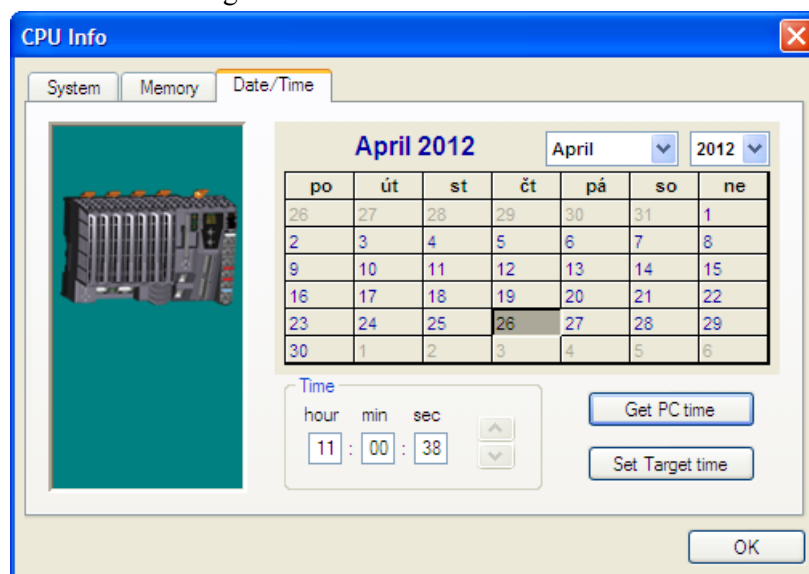
Protože PLC používá vlastní interní čas pro ukládání hodnot do záložního souboru a také jej zobrazuje ve vizualizaci, bylo potřeba, aby čas odpovídal skutečnému času. Muselo se tedy nastavit časové pásmo a automatické přepínání mezi zimním a letním časem.



Obr. 19 Nastavení systémového času.

Toto nastavení se provádí v konfiguračním souboru sysconf.br, který se nachází v okně Configuration View. Po otevření tohoto souboru se zobrazí nové okno System Software Properties. V záložce Time Synchronisation se vybere časové pásmo (GMT +01:00) a automatické přepínání letního a zimního času zaškrtnutím políčka Automatically adjust clock for daylight saving changes.

Po provedení tohoto nastavení je potřeba nastavit aktuální čas. Toto nastavení se provádí za chodu PLC v On-line režimu. V okně s hardwarovou konfigurací je potřeba stisknout pravým tlačítkem myši na hlavní modul a následně vybrat položku Online info.... Otevře se nové okno, kde se v záložce Date/Time stiskem tlačítka Get PC time získá aktuální čas z počítače a stiskem tlačítka Set Target time se čas nastaví v PLC.



Obr. 20 Nastavení data a času v PLC

5.3 Konfigurace komunikačních rozhraní a vstup/výstupních modulů

Konfigurace komunikačních rozhraní, vstupů a výstupů se provádí v souboru ArConfig.rtc. Otevřít tento soubor je možné z okna Configuration View. V tomto souboru byly nastaveny tyto parametry:

\$root -	Default gateway -	194.228.204.153
	DNS server addresses - Server 1:	158.196.133.60
		Server 2: 158.196.133.119
IF2 -	Internet address:	194.228.204.158
	Subnet mask:	255.255.255.248
IF6.ST5 -	Channel 01 - Sensor type:	PT1000
	Channel 02 - Sensor type:	PT1000
	Channel 03 - Sensor type:	PT1000
	Channel 04 - Sensor type:	PT1000
IF6.ST6 -	Channel 01 - Sensor type:	PT1000
	Channel 02 - Sensor type:	PT1000

	Channel 03 -	Sensor type:	PT1000
	Channel 04 -	Sensor type:	PT1000
IF6.ST7 -	Channel 01 -	Sensor type:	PT1000
	Channel 02 -	Sensor type:	PT1000
	Channel 03 -	Sensor type:	PT1000
	Channel 04 -	Sensor type:	PT1000
IF6.ST8 -	Channel 01 -	Sensor type:	PT1000
	Channel 02 -	Sensor type:	PT1000
	Channel 03 -	Sensor type:	PT1000
	Channel 04 -	Sensor type:	PT1000
IF6.ST9 -	Channel 01 -	Sensor type:	PT1000
	Channel 02 -	Sensor type:	PT1000
	Channel 03 -	Sensor type:	PT1000
	Channel 04 -	Sensor type:	PT1000
IF6.ST10 -	Channel 01 -	Sensor type:	PT1000
	Channel 02 -	Sensor type:	PT1000
	Channel 03 -	Sensor type:	PT1000
	Channel 04 -	Sensor type:	PT1000
IF6.ST11 -	Channel 01 -	Sensor type:	PT1000
	Channel 02 -	Sensor type:	off
	Channel 03 -	Sensor type:	off
	Channel 04 -	Sensor type:	off
IF6.ST12 -	Channel 01 -	Sensor type:	PT1000
	Channel 02 -	Sensor type:	PT1000
	Channel 03 -	Sensor type:	off
	Channel 04 -	Sensor type:	off

Všechny ostatní parametry mají ponechány defaultní nastavení. U modulů pro sériové rozhraní RS232 a RS485 se parametry nastavují v inicializačním programu obsluhy daného rozhraní.

5.4 Přiřazení vstupů a výstupů jednotlivým proměnným

Přiřazení vstupů a výstupů jednotlivým proměnným se provede dvojklikem na příslušný modul v okně Physical View. V okně Workbook se zobrazí I/O mapping, kde se jednotlivým kanálům přiřadí proměnné, do kterých se ukládají hodnoty vstupů a výstupů. U analogových modulů AT4222 se přiřadí jednotlivým kanálům proměnné datového typu integer. Teploty jsou v °C vynásobené deseti. Zde je také možné přiřadit osmibitovou proměnnou typu unsigned integer pro uložení hodnoty definující stav vstupů. Každé dva bity určují stav jednoho kanálu. U digitálních vstupů a výstupů, moduly DI6371 a DO9322, se přiřazuje jednotlivým kanálům proměnné typu boolean. u modulu CM8281 je použit analogový výstup, kterému je přiřazena proměnná typu integer.

5.5 Program PLC

Program PLC je rozdělen na několik menších programů.

- 1) RSOOneWire – vyčítání teplot z monitorovacích vrtů
- 2) CteniHodnotKalorimetru – vyčítání aktuálních měřených hodnot z kalorimetru
- 3) Rizeni_zpatecky – regulace teploty zpátečky do kogenerační jednotky
- 4) Bezpecnostni_rizeni_zpatecky – hlídání kritické teploty
- 5) Rizeni_prutoku – regulace čerpadla v hlavním okruhu.
- 6) RychlaSmycka – zachytávání rychlejších stavů
- 7) RizeniSystemu – ostatní nezařazené akce
- 8) ExportData – ukládání zálohových dat
- 9) Porucha – hlídání poruchových stavů
- 10) Visu - vizualizace

5.5.1 Vyčítání teplot z monitorovacích vrtů

Vyčítání teplot pěti monitorovacích vrtů provádí program RSOOneWire. Tento program byl do softwaru PLC implementován kolegou Tomášem Vavrlou, který tuto problematiku řešil jako diplomovou práci. Vyčtené hodnoty jsou ukládány do proměnných, které jsou dále ukládány do souboru na FTP serveru PLC a také odesílány přes Ethernet na školní server. Po domluvě byla kolegou do programu přidána obsluha proměnných NastalaChyba a ChybovaZprava pro hlášení chybových stavů. Pokud dojde v programu k nějaké chybě, zapíše se do proměnné ChybovaZprava text, vyjadřující jaká chyby nastala, a poté se nastaví proměnná NastalaChyba do jedničky.

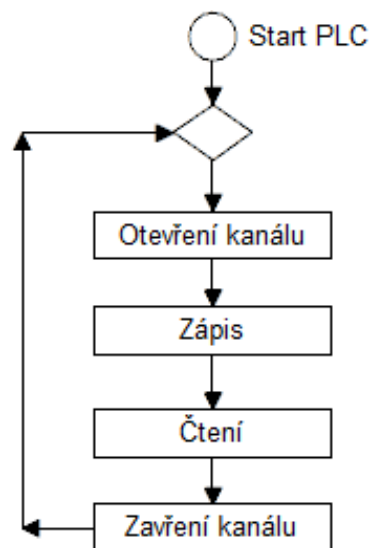
5.5.2 Čtení hodnot kalorimetru

Program CteniHodnotKalorimetru má za úkol vyčítat naměřené hodnoty z kalorimetru. Hodnoty měřené kalorimetrem jsou energie, objem, výkon, průtok, vstupní teplota, výstupní teplota a rozdíl teplot. Protože hodnota průtoku je použita pro regulaci čerpadla, byla zvolena perioda vyčítání jedna vteřina. Perioda cyklického volání programu byla nastavena na 200 ms z toho důvodu, že program nečeká na odpověď kalorimetru, ale vyšle bajtový rámec s dotazem na kalorimetr a pak se dotazuje na odpověď do té doby, než odpověď přijde.

PLC s kalorimetrem komunikuje pomocí rozhraní RS485. Pro obsluhu tohoto rozhraní je v PLC určená knihovna dvframe, obsluhující rozhraní s rámcovou komunikací jako RS232, RS422, nebo RS485. Tato knihovna zajišťuje programátorovi jednoduchou aplikační vrstvu a stará se o doplnění komunikačního rámce potřebnými údaji jako start bit, stop bit, parita, atd. Z helpu programovacího prostředí byl použit příklad kódu pro obsluhu knihovny. V knihovně jsou vytvořeny funkční bloky, které se využívají pro jednotlivé akce komunikace. Ke každému funkčnímu bloku je vytvořena struktura pro vstupní a výstupní proměnné funkčního bloku.

5.5.2.1 Obsluha rozhraní RS485

Po startu PLC se začne provádět cyklický program, ve kterém se provede otevření kanálu, zapsání, odeslání a ověření odeslání na linku RS485, čtení z linky a uzavření kanálu. Tato sekvence se opakuje s periodou jedné vteřiny.



Obr. 21 Diagram obsluhy RS485

Inicializace

Inicializace se provádí v inicializační části programu, která se provede jednou po startu PLC. V této části se provádí nastavení proměnných, určujících komunikační interface, parametry komunikace, konfigurační struktury, určující parametry vyrovnávacích pamětí, a sestavení řetězce pro požadavek na kalorimetr.

Do proměnné `string_device` se ukládá řetězec znaků určující adresu využívaného interface zařízení. Tento parametr lze zjistit v okně hardwarové konfigurace ve sloupci slot. V tomto případě se ukládá řetězec `SS1.IF1` reprezentující Sub Slot 1 a Interface 1.

Do proměnné `string_mode` se ukládá řetězec `/BD=9600 /DB=8 /PA=E /SB=1`, reprezentující nastavení těchto parametrů:

BD = 9600 – Komunikační rychlost

DB = 8 – počet datových bitů

PA = E – sudá parita

SB = 1 – stop bit

Konfigurační struktura obsahuje tyto parametry:

idle – tímto parametrem se nastavuje doba nečinnosti v počtech znaků, po které přijímač hodnotí, že je rámec kompletní. Nastavená hodnota 4 znaky

delimc – povolení oddělovacího znaku. Nastavená hodnota 0

delim – oddělovací znak. Nastavená hodnota 00

tx_cnt – výběr z osmi vyrovnávacích registrů pro odesílání. Hodnota 0
rx_cnt – výběr z osmi vyrovnávacích registrů pro příjem. Hodnota 0
tx_len – délka vyrovnávacího registru pro odesílání. Nastavená délka 10 B
rx_len – délka vyrovnávacího registru pro příjem. Nastavená délka 100 B

Sestavení požadavku na kalorimetr je vytvořeno tak, aby mohl být dodatečně upravován, například při změně adresy kalorimetru. Do řetězce `write_data` se nakopíruje pět bajtů obsahujících požadavek na kalorimetr, který se odesílá na komunikační linku. Obsah požadavku na kalorimetr viz. Princip komunikace protokolu M-Bus u zařízení CALOR 40. Část programu pro sestavení požadavku na kalorimetr je v neveřejné příloze č. IV.

Otevření komunikačního kanálu

Pro otevření komunikačního kanálu se používá funkční blok `FRM_xopen()`. Nejprve se zapíše do struktury vstup/výstupních parametrů povolení otevření kanálu. Poté se z proměnné `string_device` nakopíruje adresa zařízení, z proměnné `string_mode` se zkopíruje řetězec s parametry komunikace a ze struktury `config_struct` zkopíruje konfigurace vyrovnávacích pamětí. Poté se provede volání funkce `FRM_xopen()` se vstupními parametry uloženými ve struktuře. Po provedení funkce se ze struktury vyčte do proměnné `status_open` stav otevření a do proměnné `frm_ident` se uloží identifikační číslo kanálu. Nakonec se provede kontrolu proměnné `status_open`, zda proběhlo otevření úspěšně. Pokud se kanál otevřít nepodařilo, vypíše se do řetězce `ChybovaZprava`, že se nepodařilo otevřít kanál RS485, a nastaví se příznak `NastalaChyba`.

Využití funkčního bloku pro otevření kanálu:

```
FRM_xopen_01.enable = 1
FRM_xopen_01.device = ADR(string_device)
FRM_xopen_01.mode = ADR(string_mode)
FRM_xopen_01.config = ADR(config_struct)
FRM_xopen_01 FUB FRM_xopen()
status_open = FRM_xopen_01.status
frm_ident = FRM_xopen_01.ident
IF (status_open <> 0) THEN
    error = 1
ENDIF
```

Odeslání datového řetězce

Samotné odesílání řetězce má tři části, v každé části se používá jeden funkční blok. Nejprve je potřeba získat vyrovnávací paměť, k čemuž se využívá funkční blok `FRM_gbuf()`. Provede se opět zápis do struktury: povolení, identifikační číslo kanálu z proměnné `frm_ident`. Pak se provede volání funkce `FRM_gbuf()` pro získání paměti. Do proměnné `send_buffer` se přiřadí adresa získané vyrovnávací paměti a do proměnné `send_buffer_length` se přiřadí velikost získané paměti. Dále se do proměnné `status_gbuf` přiřadí hodnota stavu získávání paměti. Pokud

vše proběhlo v pořádku, nastaví se hodnota stavu na nulu, čímž se povolí provedení další části odeslání řetězce, tedy zapsání do vyrovnávací paměti.

Pro zapsání do vyrovnávací paměti je určen funkční blok FRM_write(). Před využitím tohoto funkčního bloku se nakopíruje na adresu získané vyrovnávací paměti řetězec určený pro odeslání, tedy požadavek na kalorimetr, který je uložený v proměnné write_data. Pak se opět provede povolení a přiřazení identifikačního čísla komunikačního kanálu. Přiřadí se adresa uloženého řetězce, tedy send_buffer a jeho délka, a pak se provede volání funkce FRM_write() pro zápis na komunikační rozhraní. Do proměnné status_write se přiřadí hodnota stavu zápisu a povolí se čtení z komunikačního kanálu. Dojde-li k chybě, nastaví se hodnota stavu zápisu na hodnotu různou od nuly. V tom případě se vypíše chyba a provede poslední část, uvolnění vyrovnávací paměti, pro které je určen funkční blok FRM_robuf(). Opět se provede povolení funkce, přiřazení identifikačního čísla komunikačního kanálu a přiřazení adresy vyrovnávací paměti s její délkou. Zavolá se funkce FRM_robuf(). Do proměnné status_robuf se přiřadí hodnota stavu uvolnění vyrovnávací paměti. Pokud je hodnota status_robuf různá od nuly, vypíše se chyba.

Program pro odeslání bajtového rámce:

```
IF (error = 0) THEN
    FRM_gbuf_01.enable = 1
    FRM_gbuf_01.ident = frm_ident
    FRM_gbuf_01 FUB FRM_gbuf()
    send_buffer = FRM_gbuf_01.buffer
    send_buffer_length = FRM_gbuf_01.buflng
    status_gbuf = FRM_gbuf_01.status
    IF (status_gbuf = 0) THEN
        memcpy(send_buffer, ADR(write_data), send_buffer_length)
        FRM_write_01.enable = 1
        FRM_write_01.ident = frm_ident
        FRM_write_01.buffer = send_buffer
        FRM_write_01.buflng = send_buffer_length
        FRM_write_01 FUB FRM_write()
        status_write = FRM_write_01.status
        start_read = 1
        IF (status_write <> 0) THEN
            FRM_robuf_01.enable = 1
            FRM_robuf_01.ident = frm_ident
            FRM_robuf_01.buffer = send_buffer
            FRM_robuf_01.buflng = send_buffer_length
            FRM_robuf_01 FUB FRM_robuf()
            status_robuf = FRM_robuf_01.status
            error = 3
            IF (status_robuf <> 0) THEN
```

```

error = 4
ENDIF
ENDIF
ELSE
error = 2
ENDIF
ENDIF
ENDIF

```

Přijetí datového řetězce

Pro čtení z komunikační linky jsou používány dva funkční bloky. FRM_read(), který se používá pro vyčtení z komunikačního rozhraní, a FRM_rbuf, uvolňující vyrovnávací paměť.

Pokud je povoleno čtení, start_read = 1, uloží se do struktury povolení čtení a identifikační číslo komunikačního kanálu. Zavoláním funkce FRM_read() se provede vyčtení dat z komunikačního rozhraní a uloží se do vyrovnávací paměti. Dále se provede uložení adresy vyrovnávací paměti do proměnné read_buffer a do proměnné read_buffer_length se uloží její velikost. Do proměnné status_read se přiřadí hodnota stavu čtení. v případě, že je hodnota stavu čtení různá od nuly, vypíše se chyba. V opačném případě se provede kopírování přijatého řetězce z vyrovnávací paměti do proměnné read_data a uvolnění vyrovnávací paměti, pomocí funkčního bloku FRM_rbuf(). Do struktury se запиše povolení funkce, identifikační číslo komunikačního kanálu a adresa vyrovnávací paměti s její velikostí. Provede se zavolání funkce, do proměnné status_rbuf se přiřadí stav uvolnění paměti, zakáže se čtení vynulováním proměnné start_read a povolí se zavření komunikačního kanálu. V případě, že dojde k chybě, hodnota status_rbuf je různá od nuly, vypíše se chybové hlášení.

Program pro přijetí bajtového rámce:

```

IF (start_read = 1) THEN
    FRM_read_01.enable = 1
    FRM_read_01.ident = frm_ident
    FRM_read_01 FUB FRM_read
    read_buffer = FRM_read_01.buffer
    read_buffer_length = FRM_read_01.buflng
    status_read = FRM_read_01.status
    IF (status_read = 0) THEN
        memcpy(ADR(read_data), read_buffer, read_buffer_length)
        FRM_rbuf_01.enable = 1
        FRM_rbuf_01.ident = frm_ident
        FRM_rbuf_01.buffer = read_buffer
        FRM_rbuf_01.buflng = read_buffer_length
        FRM_rbuf_01 FUB FRM_rbuf()
        status_rbuf = FRM_rbuf_01.status
        start_read = 0
        close = 1
        IF (status_rbuf <> 0) THEN

```

```

error = 6
ELSE
error = 0
ENDIF
ELSE
error = 5
ENDIF
ENDIF

```

Zavření komunikačního kanálu

Pokud se dokončí komunikace, provede se zavření kanálu pomocí funkčního bloku FRM_close(). Proveďte se povolení zavření kanálu, přiřadí se identifikační číslo komunikačního kanálu a zavolá se funkce FRM_close. Nakonec se vyčte stav uzavření kanálu do proměnné status_close, na základě které se pak zjišťuje, zda se podařilo kanál uzavřít. v případě, že dojde k chybě, nakopíruje se do proměnné ChybovaZprava řetězec „nepodařilo se zavřít kanál RS485“ a nastaví se příznak NastalaChyba.

Pro komunikaci je použit protokol M-Bus, pro který nemá Automation Studio žádnou podporu. Bylo tedy potřeba implementaci protokolu naprogramovat jednorázově pro tuto aplikaci. Z protokolu byla použita pouze funkce vyčtení aktuálních hodnot. Ostatní funkce, jako vyčtení archivu nebo nastavování kalorimetru, nebyly použity. Program tedy odesílá pouze jeden požadavek pro odeslání aktuálních naměřených hodnot kalorimetru v intervalu jedné vteřiny. Příchozí bajtový rámec je tedy potřeba sestavit do použitelné formy.

5.5.2.2 Sestavení dat z příchozího bajtového rámce

Přijatý bajtový rámec je uložený v osmdesáti pěti bajtovém řetězci read_data[0...84]. Významy jednotlivých bajtů jsou v tabulce, která je součástí přílohy této práce. Informace o jedné veličině nese šest po sobě jdoucích bajtů. První bajt určuje, z kolika bajtů je příslušná veličina složena, druhý bajt nese informaci o jednotkách veličiny a zbývající čtyři nesou hodnotu veličiny. Hodnoty obou energií, objemu, výkonu i průtoku jsou tedy přenášeny jako čtyřbajtová čísla datového typu integer. Teploty jsou rovněž přenášeny jako čtyřbajtová čísla, ale datového typu real. Je tedy zřejmé, že pro uložení jednotlivých veličin je potřeba složit čtyři bajty a složenou hodnotu vynásobit příslušnými jednotkami, aby byly veličiny ukládány v základních jednotkách, nebo v jejich obecných násobcích.

Kontrolní součet

Zařízení, odesílající zprávu na linku, odešle start bajt, dvakrát bajt s délkou datového řetězce a opět start bajt. Po odeslání druhého start bajtu začne přičítat jednotlivé odesílané bajty k jednobajtové proměnné. Tímto způsobem sečte celý rámec a na konec rámce odešle tento součet a pak odešle stop bajt. Tento součet neodpovídá reálnému součtu všech bajtů vlivem přetékání součtového bajtu a tím ztrátou dat, ale pro kontrolu správnosti dat je to dostačující. Zařízení, přijímající zprávu, bajtový rámec rovněž sečte stejným způsobem a porovná s hodnotou kontrolního součtu odeslaného vysílajícím zařízením. V datovém rámci odesílaném

kalorimetrem je sčítán kontrolní součet od pátého bajtu. Program pro vytvoření kontrolního součtu je v neveřejné příloze č. IV.

Složení bajtů

Pro skládání bajtů do jedné proměnné bylo nejvhodnější použití funkce `memcpy()`, která pracuje přímo s bajty a kopíruje paměť z jedné oblasti do druhé. Vstupními argumenty funkce je adresa cílové oblasti paměti, adresa zdrojové oblasti paměti a počet kopírovaných bajtů. Příklad programu je v neveřejné příloze č. IV.

Určení jednotek

Pro určení násobku jednotlivých veličin byla zvolena funkce `CASE()`, kdy se na základě hodnoty vstupního parametru vybírá, která akce se má provést. Vstupním parametrem je vždy druhý bajt ze šestice bajtů, určující jednotku veličiny. Hodnoty akcí odpovídají hodnotám, kterých příslušný bajt nabývá. V rámci akce se provede přiřazení násobku proměnné pro násobek příslušné veličiny.

Hodnoty měřených veličin jsou ukládány v obecných násobcích. Energie v gigajoulech, objem v metrech krychlových, výkon v kilowattech a průtok v metrech krychlových za hodinu. Aby nedocházelo k velké ztrátě dat, převádí se hodnoty při dělení násobkem z datového typu `integer` na typ `real`. Při dělení se zároveň vypočítává celková energie v zásobníku, odčítáním záporné energie od energie kladné. Příklad programu pro určení jednotek je v neveřejné příloze č. IV.

Zaokrouhlení teplot

Vstupní teplota, výstupní teplota a rozdíl teplot jsou posílány z kalorimetru s přesností na čtyři desetinná místa. Zadavatel požaduje hodnoty pouze s jedním desetinným místem, proto je potřeba nadbytečná desetinná místa odstranit. Odstranění se provádí vynásobením teplot deseti a převedením čísla na datový typ `integer`, čímž dojde k odstranění desetinných míst. Následně se teploty opět vydělí deseti a vznikne číslo s jedním desetinným místem.

5.5.3 Řízení zpátečky

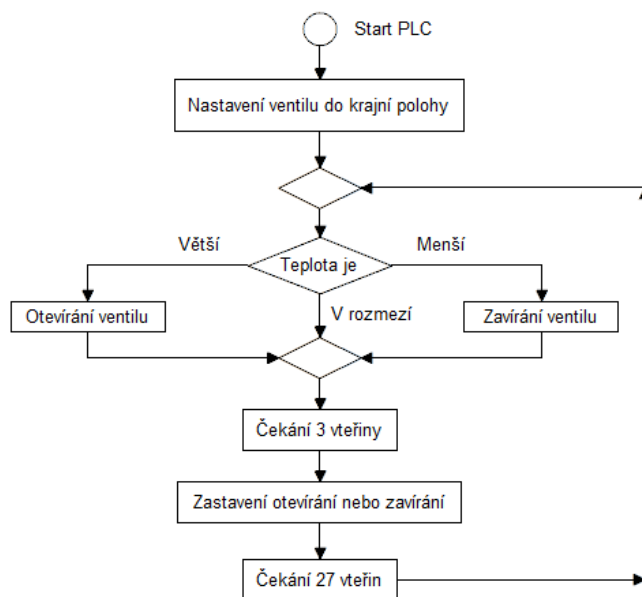
V programu `Rizeni_zpatecky` se provádí regulace teploty topného média, vracejícího se do kogenerační jednotky. Regulace se provádí mísením vracejícího se topného média s horkým topným médiem z přívodu pomocí trojcestného regulačního ventilu s elektrickým pohonem. Elektrický pohon má dva digitální vstupy Y1 a Y2. Pokud je na vstupech logická nula, pohon je v klidovém stavu. Přivede-li se na některý vstup logická jednička, pohon otáčí ventilem jedním

nebo druhým směrem. Doba přestavení pohonu z jedné krajní polohy do druhé trvá 120 sekund. Teplota topného média se snímá teplotním čidlem PT1000, instalovaným v jímce do potrubí zpátečky. Minimální teplota, na kterou je regulováno, se nastavuje ve vizualizačním prostředí.

5.5.3.1 Princip regulace

Po startu PLC se provede inicializace, kdy se vstup Y2 nastaví na logickou jedničku a vstup Y1 na logickou nulu. Pak se začne provádět cyklické volání programu s periodou jedné sekundy, ve kterém se nejprve čeká po dobu 127 period (sekund), aby se ventil jistě přestavil do krajní polohy. Čekání je realizováno inkrementací proměnné InitProdleva v každém cyklu programu pomocí funkce INC(). Pokud proměnná překročí hodnotu 127, v podmínce IF se provede povolení regulace teploty pomocí proměnné Regulace. Protože regulaci není potřeba provádět při režimu vybíjení, je inkrementace proměnné InitProdleva a aktivace regulace povolena, jen pokud je proměnná Vybíjení nastavena na hodnotu FALSE. Regulace se provádí po malých krocích, kdy se nastaví některý vstup pohonu na dobu tří sekund do logické jedničky, čímž se ventil přivře nebo pootevře, s časovými rozestupy třiceti sekund, aby se stihla teplota ustálit, z důvodu velkých časových konstant regulované soustavy. Aby se ventil neustále nepřestavoval vlivem reakce na malé odchylky, je do regulace zavedeno pásmo necitlivosti v rozmezí $\pm 2\text{ }^{\circ}\text{C}$.

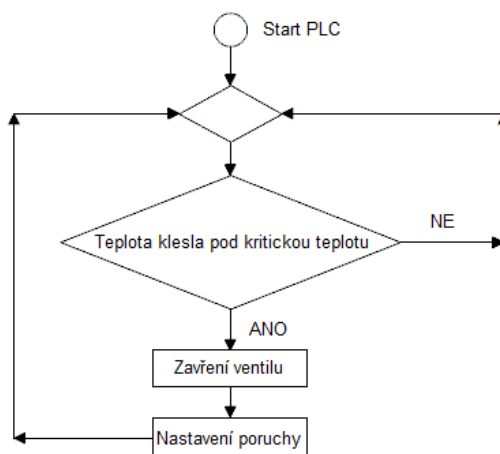
V každém cyklu programu regulace se funkcí INC() inkrementuje proměnná ProdlevaRizeniZpatecky, pomocí které je realizována třicetisekundová prodleva mezi regulacemi. Pokud tato proměnná překročí hodnotu 29, provede se vynulování proměnné ProdlevaRizeniZpatecky a spustí se jeden krok regulace. Regulace je v programu realizována podmínkami IF. Pokud je teplota zpátečky vyšší než nastavená minimální teplota zpátečky zvýšená o 2°C , provede se nastavení vstupu pohonu Y1 na logickou jedničku a vstup Y2 na logickou nulu, čímž se začne zpátečka otevírat. V opačném případě, pokud je teplota zpátečky nižší než nastavená minimální teplota zpátečky snižená o 2, provede se nastavení vstupu pohonu Y1 na logickou nulu a vstup Y2 na logickou jedničku, čímž se začne zpátečka zavírat. Následně pokud ProdlevaRizeniZpatecky překročí hodnotu 3, provede se nastavení obou vstupů pohonu na logickou nulu. Tím se zastaví otevírání, nebo zavírání zpátečky. Na obrázku (Obr. 22) je diagram regulace zpátečky.



Obr. 22 Diagram regulace zpátečky

5.5.3.2 Bezpečnostní řízení zpátečky

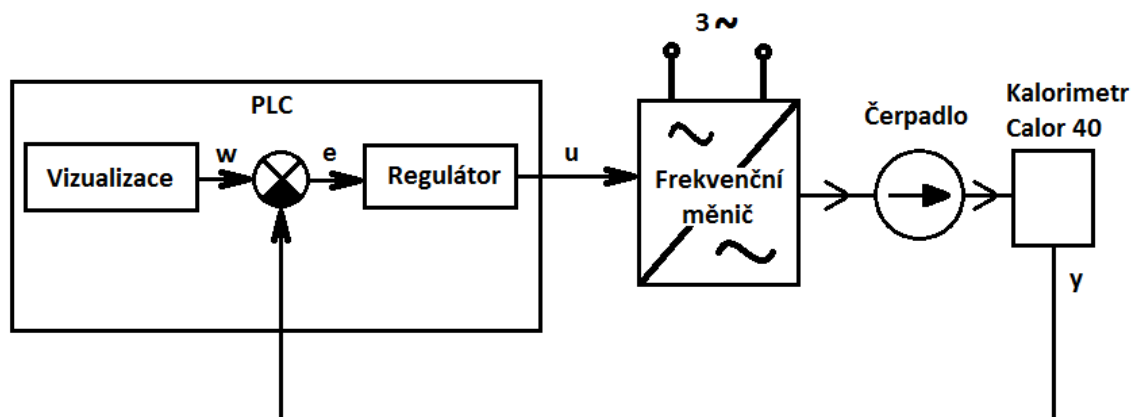
Program Bezpečnostni_řízení_zpatecky má za úkol hlídat, aby teplota topného media, vracejícího se do kogenerační jednotky, neklesla pod kritickou hodnotu. Tato hodnota se rovněž nastavuje ve vizualizaci. Hlídaní není potřeba provádět při režimu vybíjení. Program je realizován jednou podmínkou. Pokud je teplota zpátečky menší než nastavená kritická hodnota a zároveň vybíjení je vypnuto, provede se uzavření zpátečky nastavením vstupu Y2 pohonu trojcestného ventilu na logickou hodnotu jedna a vstup Y1 na logickou nulu. Vypíše se chyba, že teplota klesla pod stanovenou mez, vypne se regulace zpátečky vynulováním proměnné Regulace a proměnná PoruchaPamet se nastaví do jedničky, aby se vypnul celý systém. Tento program se provádí s periodou jedné sekundy. Na obrázku (Obr. 23) je diagram bezpečnostního řízení zpátečky



Obr. 23 Diagram bezpečnostního řízení zpátečky

5.5.4 Regulace průtoku

Program Rizeni_prutoku reguluje otáčky čerpadla. Ve vizualizaci je zadána hodnota žádaného průtoku, na kterou je programem průtok regulován. Systém je navržen jako jednoduchý zpětnovazební obvod. Na obrázku (Obr. 24 **Chyba! Nenalezen zdroj odkazů.**) je náčrt regulačního obvodu.



Obr. 24 Schéma zpětnovazebního regulačního obvodu

Pro regulaci byl navržen jednoduchý proporcionálně sumační regulátor, s periodou vzorkování jedna sekunda.

Program Rizeni_prutoku je volán s periodou jedné sekundy a dá se rozdělit na tři části. Aktivace měniče, kdy se po zapnutí čerpadla čeká patnáct sekund, než naběhne software měniče. Dále kontrola funkce čerpadla, kde se porovnává žádaná a aktuální hodnota průtoku. Pokud je rozdíl žádaného a okamžitého průtoku větší než dvacet procent žádané hodnoty déle než jednu minutu, vypne čerpadlo a pokusí se spustit čerpadlo znovu. Takto provede tři pokusy, a pokud se ani potřetí nepodaří docílit žádané hodnoty průtoku, čerpadlo se vypne a nastaví se chybové hlášení. Poslední částí je samotná proporcionálně sumační regulace.

5.5.4.1 Aktivace měniče

Napájení měniče se zapíná až při zapnutí čerpadla. Měnič má svůj vlastní software a tedy chvíli trvá, než se měnič dostane do pohotovostního režimu. Čekání je realizováno tak, že po aktivaci čerpadla začne program inkrementovat proměnnou ProdlevaAktivaceMenice. Pokud tato proměnná překročí hodnotu devatenáct, nastaví se výstup pro aktivaci měniče na hodnotu jedna a vynulují se proměnné Integracni, IntegracniPomocna a ProdlevaKontrolyCerpada.

5.5.4.2 Kontrola funkce čerpadla

Kontrola funkce čerpadla se provádí proto, aby nedošlo k jeho přetížení, například špatným přepnutím klapky v topném okruhu, a pro vyhodnocení chyby, pokud by se čerpadlo nerozběhlo, kupříkladu vyskočením proudové ochrany na proudovém chrániči.

Pokud je absolutní hodnota rozdílu zadaného průtoku a okamžitého průtoku větší než dvacet procent žádaného průtoku, provede se inkrementace proměnné ProdlevaKontrolyCerpada. Když tato proměnná překročí hodnotu 59, což odpovídá jedné minutě, inkrementuje se proměnná PokusCerpadlo_1, AktivaceMenice se nastaví na nulu. Jestliže proměnná PokusCerpadlo_1 překročí hodnotu dvě, aktivuje se hlášení poruchy, čímž se vypne celý systém. Ale pokud je absolutní hodnota rozdílu zadaného průtoku a okamžitého průtoku menší než dvacet procent žádaného průtoku dříve, než proměnná ProdlevaKontrolyCerpada překročí hodnotu 59, provede se vynulování proměnné ProdlevaKontrolyCerpada a PokusCerpadlo_1.

5.5.4.3 Proporcionálně sumační regulace

U regulačního obvodu nebylo možné vypočítat žádné časové konstanty přechodové charakteristiky, proto byla proporcionální i integrační konstanta určena experimentálně. Proporcionální regulace se vypočítává jako rozdíl žádané hodnoty průtoku a aktuální hodnoty průtoku vynásobené proporcionální konstantou. Integrační regulace se vypočítává sčítáním rozdílu žádané a okamžité hodnoty průtoku vynásobené integrační konstantou. Tyto regulace se následně sčítají a přiřazují do proměnné RegulaceCerpada_1. U integrační regulace a celkové regulace se hlídá, aby nedošlo k přetečení. Pokud dojde k přetečení integrační proměnné směrem nahoru, změní se hodnota z 32767 na hodnotu -32767. Pokud přeteče hodnota do záporných hodnot, změní se hodnota z -32767 na 32767. Proto je ukládána hodnota integrační regulace z předešlého kroku. Tato hodnota je následně porovnávána s aktuální hodnotou integrační regulace, a pokud je rozdíl větší než 6000, ponechá se pro regulaci hodnota integrační regulace z předešlého kroku. U proměnné RegulaceCerpada_1 je hlídáno, aby se do této proměnné přiřazovala hodnota větší než nula. Dojde-li k přetečení proměnné AkcniVelicina nahoru nebo dolů, přejde hodnota do záporných hodnot. V tom případě se zjistí, zda je hodnota RegulaceCerpada_1 z předešlého kroku regulace blíže nule, nebo maximální hodnotě, tedy 32767 a podle toho se přiřadí RegulaceCerpada_1. Na obrázku (Obr. 25) je diagram regulace čerpadla.

Příklad programu pro PS regulaci:

```

Integracni = INT(Integracni + ((ZadanyPrutok - Prutok_m3) * I_slozka))
IF ABS(Integracni - IntegracniPomocna) > 6000 THEN
    Integracni = IntegracniPomocna
ENDIF
IntegracniPomocna = Integracni
Proporcionalni = INT((ZadanyPrutok - Prutok_m3) * P_slozka)
AkcniVelicina = (Integracni + Proporcionalni)
IF 0 < AkcniVelicina THEN
    RegulaceCerpadlo_1 = AkcniVelicina
ELSE
    IF RegulaceCerpadlo_1 < 15000 THEN
        RegulaceCerpadlo_1 = 0
    ENDIF

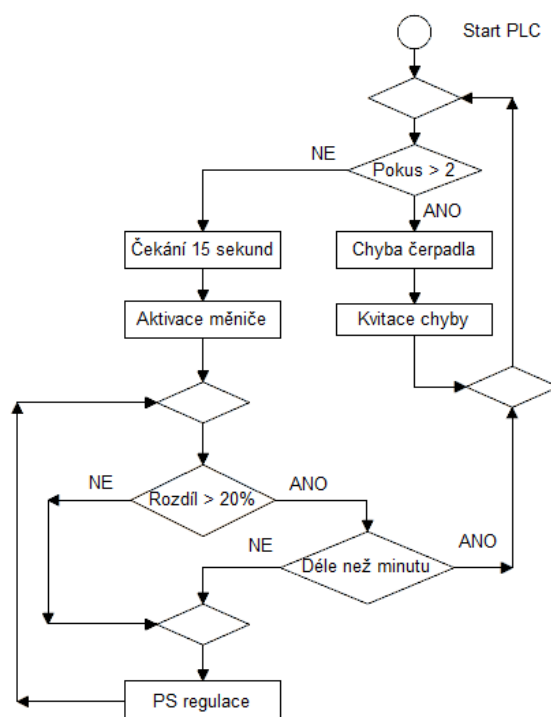
```

```

IF RegulaceCerpadlo_1 > 15000 THEN
    RegulaceCerpadlo_1 = 32767
ENDIF
ENDIF

```

Diagram regulace čerpadla



Obr. 25 Diagram regulace čerpadla

5.5.5 Rychlá smyčka

Program RychlaSmycka má za úkol zachytávat rychlejší stavy, které by se nemusely zaznamenat v ostatních pomalejších programech. Volání tohoto programu je nastaveno na periodu 50 ms. Je zde zaznamenávání stisku tlačítka kvitování poruchy. Program, zabývající se poruchami, je volán s periodou jedné sekundy. Pokud by obsluha systému stiskla toto tlačítko na dobu kratší než jedna sekunda, mohlo by se stát, že program tento stav nezaregistruje. Proto se program RychlaSmycka dotazuje, zda je na vstupu logická hodnota jedna, tedy tlačítko je stisknuto. Pokud ano, nastaví se pomocná proměnná KvitovaniPoruchyPomocna do jedničky. Této pomocné proměnné se využívá také ve vizualizaci pro kvitování poruchy, protože hodnotu proměnné, použité pro vstup není možné nastavovat jiným způsobem než nastavením vstupu. Pomocná proměnná je pak použita pro provedení akce v programu Porucha, kde je tato proměnná i nulována. Dále jsou v programu podmínky pro nastavení proměnných TlakMinimálníNabijeciPomocna a TlakMinimálníChlazeníPomocna. Tyto proměnné se rovněž nastavují na základě vstupů, ale zde není až tak podstatné aby nedošlo k nezaregistrování stavu, protože tyto vstupy mají pomalé změny, ale je potřeba, aby při poklesu některého tlaku zůstal tento stav zaznamenán pro operátora, i kdyby došlo opět ke zvýšení tlaku.

Příklad kódu:

```
IF KvitovaniPoruchy = TRUE THEN
    KvitovaniPoruchyPomocna = TRUE
ENDIF
```

5.5.6 Řízení systému

V této části programu jsou ostatní nezařazené části řízení jako změny nastavení při přepínání režimů nabíjení a vybíjení, regulace ventilátorů a převody naměřených teplot na reálné hodnoty.

Přepínání režimů

Při změně stavu proměnné Vybijeni z hodnoty FALSE na hodnotu TRUE se provede spuštění oběhového čerpadla chladicího okruhu a zapne se regulace ventilátoru. V opačném případě, kdy se změní stav proměnné Vybijeni z hodnoty TRUE na hodnotu FALSE, se provede zastavení oběhového čerpadla chladicího okruhu, vypnutí všech ventilátorů a jejich regulace. Pro sledování změny hodnoty proměnné Vybijeni se využívá proměnné VybijeniPamet, do které se ukládá předchozí hodnota proměnné Vybijeni.

Program reagující na změnu režimu:

```
IF Vybijeni = TRUE THEN
    IF VybijeniPamet = FALSE THEN
        VybijeniPamet = TRUE
        ChodObehoveCerpadlo = TRUE
        AutomatickeRizeniVentilatoru = TRUE
```

```

ENDIF
ENDIF
IF Vybijeni = FALSE THEN
    IF VybijeniPamet = TRUE THEN
        VybijeniPamet = FALSE
        ChodObecneCerpadlo = FALSE
        AutomatickeRizeniVentilatoru = FALSE
        ChodVentilatoru_1 = FALSE
        ChodVentilatoru_2 = FALSE
        ChodVentilatoru_3 = FALSE
    ENDIF
ENDIF
ENDIF

```

Řízení ventilátorů

Pro řízení rychlosti vybíjení se provádí řízení ventilátorů. V chladicím okruhu jsou instalovány tři chladiče s ventilátory. Na každém chladiči se spouští ventilátory zvlášť, čímž se dá počtem spuštěných ventilátorů měnit množství odevzdaného tepla do ovzduší.

Řízení ventilátorů se provádí na základě rozdílu teplot mezi vstupem do chladičů a výstupem z nich. Požadovaný rozdíl se nastavuje ve vizualizaci. Program funguje tak, že porovnává skutečný rozdíl teplot s požadovaným. Pokud je skutečný rozdíl menší než žádaný, provede se zapnutí ventilátorů prvního chladiče, ale pokud měl již první chladič ventilátory spuštěny, spustí se ventilátory na druhém chladiči. Pokud už jsou zapnuty i ventilátory na druhém chladiči, spustí se ventilátory na třetím chladiči. V opačném případě, to znamená, že je rozdíl teplot větší než požadovaný, provede se vypnutí ventilátorů prvního chladiče. V případě, že již byly ventilátory prvního chladiče vypnuty, vypnou se ventilátory druhého chladiče. Pokud už byly vypnuty i tyto ventilátory, vypnou se ventilátory třetího chladiče. Mezi každým spuštěním nebo vypnutím ventilátorů dalšího chladiče je minutová prodleva, pro ustálení rozdílu teplot. Aby nedocházelo k periodickému kmitání rozdílu teplot okolo žádané hodnoty zapínáním a vypínáním ventilátorů, je v řízení zavedeno pásmo necitlivosti ± 2 °C. Toto pásmo bylo určeno jen na základě subjektivního odhadu, protože k chladičům ani výměníku tepla nebyl uveden žádný výkonový propočet nebo měření. Proto bude možná nutné po spuštění režimu vybíjení toto pásmo necitlivosti upravit.

Program řízení ventilátorů:

```

IF AutomatickeRizeniVentilatoru = TRUE THEN
    IF Vybijeni = TRUE THEN
        INC(ProdlevaRizeniVentilatoru)
        IF ProdlevaRizeniVentilatoru > 59 THEN
            ProdlevaRizeniVentilatoru = 0
            IF (TeplotaVystupSekundar_REAL -
                TeplotaZpateckaSekundar_REAL) <
                (RozdilTeplotSekundar - 2) THEN
                IF ChodVentilatoru_1 = TRUE THEN

```

```

IF ChodVentilatoru_2 = TRUE THEN
    ChodVentilatoru_3 = TRUE
ELSE    ChodVentilatoru_2 = TRUE
ENDIF

ELSE    ChodVentilatoru_1 = TRUE
ENDIF

ENDIF

IF (TeplotaVystupSekundar_REAL -
- TeplotaZpateckaSekundar_REAL) >
> (RozdilTeplotSekundar + 2) THEN
    IF ChodVentilatoru_1 = FALSE THEN
        IF ChodVentilatoru_2 = FALSE THEN
            ChodVentilatoru_3 = FALSE
        ELSE    ChodVentilatoru_2 = FALSE
        ENDIF
    ELSE    ChodVentilatoru_1 = FALSE
    ENDIF
ENDIF

ENDIF

ENDIF

ENDIF

```

5.5.7 Vytváření zálohových souborů

Program ExportData vytváří soubory na FTP serveru v PLC a následně do nich zapisuje. Tyto soubory jsou určeny pro zálohování dat pro případ, že by došlo k výpadku komunikace. První částí programu je vytváření souborů pro ukládání měřených dat. Tato data jsou ukládána v PLC v pětiminutových intervalech. Na serveru, umístěném v areálu Vysoké školy báňské, se data ukládají v minutových intervalech, ale z důvodu úspory paměti to pro dodání dat při výpadku postačuje. Tato část byla vytvořena správcem školního serveru Ing. Tomášem Vavrlou. Součástí vypracování této diplomové práce je jen implementace této části do projektu a zařazení výpisu chybových hlášení do této části. Druhou částí je zapisování vzniklých chybových hlášení do souboru Error.csv a přidání hlášení do řetězce, zobrazovaného ve vizualizaci. Chybová hlášení vznikají v jednotlivých programech při výskytu dané chyby. Vyhodnotí-li některý program chybu, například klesne-li teplota média vracejícího se do kogenerační jednotky pod stanovenou mez, nakopíruje příslušný program do řetězce ChybovaZprava text, například „Teplota media, vracejícího se do kogenerační jednotky, klesla pod stanovenou minimální hodnotu.“ a nastaví proměnnou NastalaChyba na hodnotu TRUE. Tímto způsobem je zajištěn zápis jakékoli chyby a jednoduché přidání nového chybového hlášení.

Příklad části programu pro přiřazení chybového hlášení:

```

strcpy(ADR(ChybovaZprava), "Teplota media, vracejícího se do
kogenerační jednotky, klesla pod stanovenou minimální hodnotu.")
NastalaChyba = TRUE

```

Soubor s příponou csv, je obecný tabulkový soubor, ve kterém platí tato pravidla. Oddělení sloupců se provádí čárkou, středníkem, nebo tabulátorem. Posun na nový řádek se provádí klasickým odřádkováním klávesou Enter. Enter v textu vkládá dvojici znaků CR (znak pro návrat na začátek řádku) a LF (znak pro posun na další řádek).

Zápis chybových hlášení má osm částí, ve kterých se provádí volání funkčních bloků pro práci se souborem. Tyto funkční bloky mají své datové struktury pro svá vstupní a výstupní data. Výběr těchto částí programu je realizován funkcí CASE, která na základě hodnoty proměnné Step2, nastavena po startu PLC na hodnotu 0, vybírá, která část programu se má provádět. První částí je akce 0 (action 0), ve které se provádí testování, zda nastala chyba a zároveň zda je baterie dobíjena, tedy že je PLC napájeno ze sítě a ne z UPS. Pokud tato podmínka splněna není, nestane se nic a v dalším volání programu se opět testuje tuto podmínku. Pokud však podmínka splněna je, nastaví se proměnná Step2 na hodnotu jedna a vynuluje se příznak chyby, proměnná NastalaChyba. Dále se provede získání aktuálního data a času pomocí funkčního bloku DTGetTime(), který vloží do vlastní struktury DTGetTime_1 pro vstup/výstupní data. Z této struktury se pomocí funkce DT_TO_DTStructure() převedou datum a čas do struktury, ve které jsou uloženy v jednotlivých proměnných hodnoty roku, měsíce, dne, hodiny, minuty a sekundy. Z této struktury se pak jednotlivé hodnoty převádějí pomocí funkce itoa() na řetězec znaků a ukládají se do řetězců Year, Month, Day, Hour, Minute a Second. Z těchto řetězců se složí název souboru. Funkcí strcpy() se do řetězce ErrorFileName zkopíruje řetězec „Error_“ a za něj se nakopírují pomocí funkce strcat() řetězec Year, „_“, Month a „.csv“. Tím vznikne název souboru například ve tvaru Error_2012_4.csv, kde 2012 je rok a 4 je měsíc vytvoření souboru.

V novém volání programu se na základě proměnné Step2, která má nyní hodnotu 1, přejde na další část programu action 1. V této části se provádí pomocí funkčního bloku FileOpen() otevření souboru s chybovými hláškami. Do vstup/výstupní struktury tohoto funkčního bloku se запиše povolení otevření souboru, adresa proměnné NameDevice, kde je uložen název paměťového zařízení, dále se запиše adresa názvu souboru, proměnná ErrorFileName, a mód otevření souboru pro čtení a zápis, klíčové slovo FILE_RW. Posledním krokem pro otevření je zavolání funkce FileOpen(). Po zavolání funkce se do proměnné FileIdent2 запиše identifikační číslo souboru a do proměnné FileStatus2 se přiřadí hodnota stavu souboru. Pokud se hodnota stavu souboru nastaví na 20708, znamená to, že soubor neexistuje. v tom případě se nastaví proměnná Step2 na hodnotu 2, čímž se přejde na část programu pro vytvoření souboru. Nastaví-li se stav souboru na hodnotu nula, soubor existuje a je otevřen, a v tom případě se nastaví Proměnná Step2 na hodnotu 5, čímž se přejde na zápis do souboru. Nastaví-li se stav souboru na jinou hodnotu, vypíše se chybové hlášení a přejde se na akci 7, kde je zápis do řetězce pro výpis chyb ve vizualizaci.

Neexistoval-li soubor, přechází se na třetí část programu, vytvoření souboru. v této části se využívá funkčního bloku FileCreate(). Jeho vstupními parametry jsou povolení funkčního bloku, adresa jména paměťového zařízení a adresa názvu souboru. Po zavolání funkčního bloku

se nakopíruje do proměnných FileIdent2 a FileStatus2 identifikační číslo souboru a stav. Pokud je hodnota FileStatus2 rovna nule, přejde se na vytvoření hlavičky souboru a vynuluje se proměnná FileSpacing1. V případě že je stav různý od nuly, vypíše se chyba a přejde se na akci 7.

Čtvrtou částí je vytvoření hlavičky souboru. Vytvořená hlavička se ukládá do datové struktury obsahující tři řetězce znaků. První dva řetězce označují hlavičky sloupců Čas a Chybová zpráva, v každém řetězci je na konci středník pro oddělení sloupců, a v třetím řetězci Enter, je uložena dvojice znaků pro odřádkování. Poté se přejde na další krok, zápis hlavičky do souboru.

Program:

```
strcpy(ADR(FileHeader1.Cas), "Cas;")
strcpy(ADR(FileHeader1.ChybovaZprava), "Chybová zpráva;")
strcpy(ADR(FileHeader1.Enter), ADR(Enter))
```

Pátou částí je zápis hlavičky do souboru pomocí funkčního bloku FileWrite(). Vstupními parametry jsou povolení zápisu, identifikační číslo souboru, offset, který má hodnotu délky hlavičky vynásobené řádkováním, adresa proměnné, ve které je uložena hlavička souboru, a její velikost. Po zavolání funkčního bloku se testuje stav zápisu. Pokud vše proběhne v pořádku, nastaví se na hodnotu 0 a v tom případě se inkrementuje proměnná řádkování a přejde se na sestavení zapisované chybové zprávy a její zápis. V opačném případě, kdy je stav různý od nuly, se provede vypsání chybového hlášení a přejde se na akci 7.

Příklad programu:

```
FileWrite3.enable = 1
FileWrite3.ident = FileIdent2
FileWrite3.offset = SizeHeader1 * FileSpacing1
FileWrite3.pSrc = ADR(FileHeader1)
FileWrite3.len = SIZEOF(FileHeader1)
FileWrite3 FUB FileWrite()
FileStatus2 = FileWrite3.status
IF (FileStatus2 = 0) THEN
    Step2 = 5
    INC(FileSpacing1)
ELSE
    IF (FileStatus2 <> 65535) THEN
        FileErrorLevel = 27
        Step2 = 7
    ENDIF
ENDIF
```

Akcí 5 je vytvoření chybové hlášky a zapsání do souboru. Nejprve je potřeba sestavit datum a čas do řetězce FileTime. U každé hodnoty, den, měsíc, rok, hodina, minuta a sekunda se testuje, zda je větší než deset, a pokud není, vloží se nejprve 0 a poté se vloží příslušný

řetězec, například Day. Pak se vloží tečka a pokračuje se další hodnotou z data a času. Mezi rok a hodinu se místo tečky vkládá mezerník a na konec řetězce se vloží středník. K tomuto skládání se používá funkce strcat(). Složený řetězec se následně vloží do skládaného řetězce FileData1 pro zápis do souboru. Za řetězec s datem a časem se vloží řetězec ChybovaZprava, kde je uložen text příslušné chyby, která právě nastala. Nakonec se vloží dva znaky pro odřádkování a jeden prázdný z proměnné Enter. Pro toto poskládání se rovněž používá funkce strcat(). Dále se pokračuje zapsáním do souboru, což se provádí stejně jako zápis hlavičky do souboru vysvětlené v akci 4 s tím rozdílem, že adresa se zapisovanými daty není FileHeader1, ale FileData1. Pokud proběhne zápis úspěšně, přejde se na akci 7, ve které se provede uzavření souboru.

Příklad části programu pro skládání chybové zprávy:

```
IF (FileDTStructure.day < 10) THEN
    strcpy(ADR(FileTime), "0")
    strcat(ADR(FileTime), ADR(Day))
ELSE
    strcpy(ADR(FileTime), ADR(Day))
ENDIF

strcat(ADR(FileTime), ".")
```

V části programu action 6, se pomocí funkčního bloku FileClose() provede zavření souboru. Vstupními parametry jsou povolení zavření souboru a identifikační číslo souboru. Po zavolání funkčního bloku se rovněž provede testování stavu uzavření souboru. Pokud je stav roven nule, vše proběhlo v pořádku a přejde se na další krok programu. Je-li hodnota rovna 65535, čeká se na dokončení zavírání souboru. V případě že se hodnota nerovná nule ani 65535, vypíše chyba a rovněž se přejde na další krok programu.

Příklad programu pro uzavření souboru:

```
FileClose3.enable = TRUE
FileClose3.ident = FileIdent2
FileClose3 FUB FileClose()

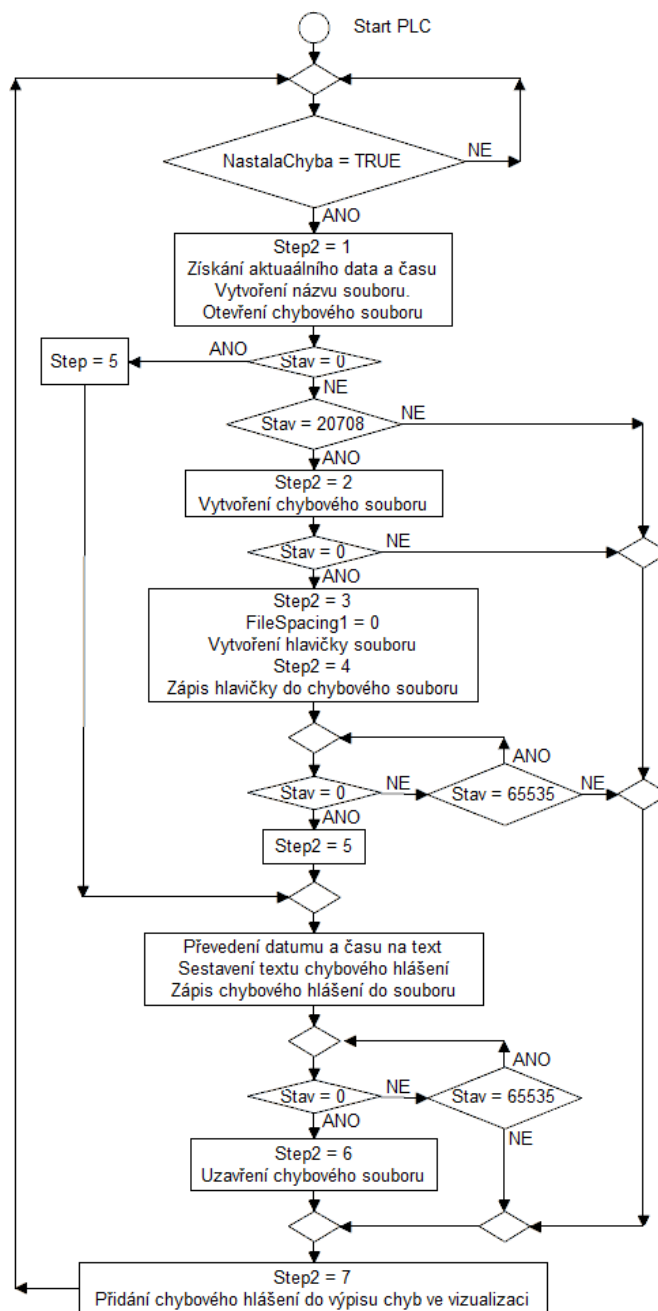
FileStatus2 = FileClose3.status
IF (FileStatus2 = 0) THEN
    Step2 = 7
ELSE
    IF (FileStatus2 <> 65535) THEN
        FileErrorLevel = 29
        Step2 = 7
    ENDIF
ENDIF
```

V následující části programu action 7 se provede přidání aktuální chybové zprávy na začátek řetězce VypisChyb zobrazovaného ve vizualizaci. Pomocí funkce strcpy() se zkopíruje řetězec VypisChyb do řetězce VypisChybPam. Následně se chybová zpráva z řetězce FileData1 nakopíruje do řetězce VypisChyb a pak se funkcí strcat() připojí za stávající text v řetězci VypisChyb původní text z řetězce VypisChybPam. Tím se docílí, že původní text se posune a nová chybová hláška se dostane na začátek pro větší přehlednost. Tímto je zápis nové chyby dokončen a proměnná pro výběr části programu Step2 se nastaví na nulu, čímž se čeká na opětovné vyvolání zápisu chyby nastavením proměnné NastalaChyba na hodnotu TRUE.

Příklad programu pro zkopírování chybového hlášení na začátek řetězce:

```
strcpy(ADR(VypisChybPam), ADR(VypisChyb))  
strcpy(ADR(VypisChyb), ADR(FileData1))  
strcat(ADR(VypisChyb), ADR(VypisChybPam))
```

Diagram zápisu do chybového souboru:



Obr. 26 Diagram zápisu do chybového souboru

5.5.8 Program obsluhy poruch

Program Porucha je volán s periodou jedné sekundy a má za úkol provést potřebná opatření v případě, že dojde k nějaké chybě. Dále provádí potřebné akce při stisku tlačítka kvitování poruchy a vyhodnocuje poruchy teplotních čidel PT1000.

Vyhodnocení poruchy teplotních čidel PT1000

Moduly AT4222 podporují vyhodnocování chyb teplotních čidel na základě jejich elektrického odporu. Každý modul pak prezentuje chyby v osmibitovém čísle, kde každé dva bity prezentují chybu jednoho teplotního čidla, které jsou schopny zaznamenat čtyři stavy.

- 00 – teplotní čidlo pracuje správně
- 01 – odpor teplotního čidla klesl pod povolenou mez
- 10 – odpor teplotního čidla stoupl nad povolenou mez
- 11 – teplotní čidlo má přerušené vedení

Program je tedy vytvořen tak, že vyhodnocuje jednotlivé dvojice bitů a na základě jejich stavu vypisuje chybová hlášení. V každém cyklu programu se provádí vyhodnocení jednoho teplotního čidla. Proměnná *n* se v každém cyklu inkrementuje a na základě její hodnoty je testováno příslušné teplotní čidlo. Celou operaci vyhodnocení jednoho teplotního čidla lze rozdělit na čtyři části. První částí je přiřazení chybové proměnné *ErrorTeplotniModul_x* z příslušného modulu proměnné *Pracovni*, se kterou se posléze pracuje. Toto přiřazení se provádí na základě hodnoty proměnné *n*. Pokud je hodnota proměnné *n* v rozmezí 0 až 3, přiřazuje se proměnné *Pracovni* chybovou hodnotu z prvního teplotního modulu. Je-li hodnota proměnné *n* v rozmezí 4 až 7 přiřadí se chybová hodnota druhého teplotního modulu. Takto se po čtyřech pokračuje až po osmý teplotní modul. Do proměnné *Modul* se přiřazuje pořadí aktuálního modulu.

Příklad programu:

```
IF (0 <= n) AND (n < 4) THEN
    Pracovni = ErrorTepotniModul_1
    Modul = 1
ELSE IF (4 <= n) AND (n < 8) THEN
    Pracovni = ErrorTepotniModul_2
    Modul = 2
:
:
:
ELSE IF (28 <= n) AND (n < 32) THEN
    Pracovni = ErrorTepotniModul_8
    Modul = 8
ENDIF
```

V dalším kroku se určuje, zda došlo na aktuálně testovaném modulu k chybě některého čidla. Pokud k nějaké chybě došlo, je hodnota proměnné *Pracovni* větší než nula. V tom případě se do řetězce *Cidlo* vloží text popisující umístění teplotního čidla. U prvních 24 čidel se tento text skládá. Matematickým výpočtem se určí čísla vrtů a zda je čidlo umístěno na jejich vstupu, mezi vrty, nebo na jejich výstupu. Pro určení čísel vrtů se nejprve zjistí, zda je *n* větší než 11, pokud ano, tak se do proměnné *PomocnaProNazevVrtu* vloží hodnota 4, jinak se nastaví na

hodnotu 0. Poté se do této proměnné uloží $n/3 + 1$ + její původní hodnota. Pak je převedena na text a ten je vložen do řetězce Vrty. Dále je k proměnné PomocnaProNazevVrtu přičtena hodnota 4 a převedena na text. Tento text je společně s podtržítkem vložen za text v řetězci Vrty. Tímto postupem vznikne text například ve tvaru „9_13“. Vydělením čísla n třemi, následným zaokrouhlením a opětovným vynásobením třemi se získá násobek tří. Po jeho odečtení od čísla n získáme číslo 0, 1, nebo 2, což odpovídá umístění na vstupu do vrtů, mezi vrty, nebo jejich výstupu. Podle toho je přiřazen do řetězce Cidlo text „na vstupu V“, „mezi vrty V“, nebo „na výstupu V“. Za tento text je pak vložen text z řetězce Vrty, čímž vznikne například text „na vstupu V9_13“. u teplotních čidel 24 – 31 se vkládá do řetězce Cidlo přímo text popisující umístění teplotního čidla.

Část programu pro složení textu popisujícího umístění teplotního čidla:

```

IF n < 24 THEN
    PoziceBitu = n/3
    IF n > 11 THEN
        PomocnaProNazevVrtu = 4
    ELSE
        PomocnaProNazevVrtu = 0
    ENDIF
    PomocnaProNazevVrtu = USINT(PoziceBitu)+1+PomocnaProNazevVrtu
    itoa(PomocnaProNazevVrtu, ADR(Cislo))
    strcpy(ADR(Vrty), ADR(Cislo))
    PomocnaProNazevVrtu = PomocnaProNazevVrtu + 4
    itoa(PomocnaProNazevVrtu, ADR(Cislo))
    strcat(ADR(Vrty), "-")
    strcat(ADR(Vrty), ADR(Cislo))
    IF ((n - (3*PoziceBitu))) = 0 THEN
        strcpy(ADR(Cidlo), "na vstupu V")
        strcat(ADR(Cidlo), ADR(Vrty))
    ELSE IF ((n - (3*PoziceBitu))) = 1 THEN
        strcpy(ADR(Cidlo), "mezi vrty V")
        strcat(ADR(Cidlo), ADR(Vrty))
    ELSE IF ((n - (3*PoziceBitu))) = 2 THEN
        strcpy(ADR(Cidlo), "na výstupu V")
        strcat(ADR(Cidlo), ADR(Vrty))
    ENDIF
ELSE IF n = 24 THEN
    strcpy(ADR(Cidlo), "na zpátečce do kogenerační jednotky")
ELSE IF n = 25 THEN
    strcpy(ADR(Cidlo), "za anuloidem")
ELSE IF n = 26 THEN
    strcpy(ADR(Cidlo), "na vstupu do rozdělovače")
ELSE IF n = 27 THEN

```

```

        strcpy(ADR(Cidlo), "na výstupu ze sběrače")
ELSE IF n = 28 THEN
    strcpy(ADR(Cidlo), "na vstupu do sekundárního okruhu")
ELSE IF n = 29 THEN
    strcpy(ADR(Cidlo), "na zpátečce ze sekundárního okruhu")
ELSE IF n = 30 THEN
    strcpy(ADR(Cidlo), "")
ELSE IF n = 31 THEN
    strcpy(ADR(Cidlo), "")
ENDIF

```

Další částí programu se prováděno vyhodnocení k jaké chybě na testovaném teplotním čidle došlo. Toto vyhodnocení se provádí pomocí funkce SHR(). Do proměnné ShiftR se přiřadí hodnota proměnné Pracovni bitově posunutá o $2 \cdot (n - 4 \cdot (\text{Modul} - 1))$ bitů a bitově vynásobené proměnnou Maska, kde je uložena hodnota 3. Posun nabývá hodnot 0, 2, 4 a 6. Proměnná ShiftR nabývá hodnot 0, 1, 2 nebo 3. Na základě této hodnoty je pak funkcí CASE vybírán text, který se nakopíruje do řetězce ChybovaZprava. Chybové hlášení může vypadat například takto: „Odpor teplotního čidla na vstupu V9_13 klesl pod povolenou mez.“

Část programu vyhodnocující chybu teplotního čidla:

```

ShiftR = (SHR(Pracovni, (2*(n - 4*(Modul - 1))))) AND Maska
CASE ShiftR OF
ACTION 1:
    strcpy(ADR(ChybovaZprava), "Odpor teplotního čidla ")
    strcat(ADR(ChybovaZprava), ADR(Cidlo))
    strcat(ADR(ChybovaZprava), " klesl pod povolenou mez.")
    PovolitZapisChyby[n] = TRUE
ENDACTION
ACTION 2:
    strcpy(ADR(ChybovaZprava), "Odpor teplotního čidla ")
    strcat(ADR(ChybovaZprava), ADR(Cidlo))
    strcat(ADR(ChybovaZprava), " stoupl nad povolenou mez.")
    PovolitZapisChyby[n] = TRUE
ENDACTION
ACTION 3:
    strcpy(ADR(ChybovaZprava), "Teplotní čidlo ")
    strcat(ADR(ChybovaZprava), ADR(Cidlo))
    strcat(ADR(ChybovaZprava), " má přerušené vedení.")
    PovolitZapisChyby[n] = TRUE
ENDACTION
ENDCASE

```

Poslední částí vyhodnocování chyb teplotních čidel je opatření, aby nedocházelo k vypisování vzniklé poruchy v každém cyklu programu, ale jen jednou při vzniku poruchy.

A nastane-li porucha na teplotním čidle měřící teplotu media vracejícího se do kogenerační jednotky, nastaví se porucha systému a systém se zastaví. Pro zajištění jednoho vypsání poruchy jsou vytvořeny dvě paměťová pole o velikostech 32 hodnot. Při vzniku poruchy se na n-tou pozici v poli PovolitZapisChyby zapíše logická jednička. Toto pole se následně porovnává s polem PovolitZapisChybyPam. Pokud se tato pole nerovnají, povolí se zápis poruchy a do pole PovolitZapisChybyPam se nakopíruje hodnoty pole PovolitZapisChyby. V dalších cyklech programu se již pole rovnají, a tedy nedojde k zápisu poruchy do doby, kdy vznikne nová porucha. Po odstranění poruchy musí provést provozovatel systému kvitování poruchy stiskem tlačítka Kvitování poruchy. Tímto stiskem se provede vynulování obou polí a tím může na opraveném čidle po vzniku nové poruchy dojít k zápisu poruchy. Dojde-li k povolení zápisu poruchy a proměnná n má hodnotu 24, což odpovídá testování teplotního čidla na zpátečce do kogenerační jednotky, nastaví se proměnná PoruchaPamet na hodnotu jedna, a tím se zastaví celý systém vytápění. V ostatních případech se nastaví na hodnotu jedna proměnná Varovani.

Část programu pro povolení jednoho zápisu poruchy:

```
IF NOT (PovolitZapisChyby[n] = PovolitZapisChybyPam[n]) THEN
    PovolitZapisChybyPam[n] = PovolitZapisChyby[n]
    NastalaChyba = TRUE
    IF n = 24 THEN
        PoruchaPamet = TRUE
    ELSE
        Varovani = TRUE
    ENDIF
ENDIF
```

Vyhodnocování poruchy tlaků a ventilátorů

Tato část provádí testování zda došlo k poklesu tlaku pod stanovenou mez v některém okruhu. Při poklesu tlaku v hlavním okruhu se nastaví proměnná TlakMinimalniNabijeciPomocna na hodnotu jedna. V tom případě se provede zápis chybového hlášení a proměnná PoruchaPamet se nastaví na hodnotu jedna. Při poklesu tlaku v chladicím okruhu se nastaví proměnná TlakMinimalniChlazeníPomocna na hodnotu jedna. Pokud je systém v režimu vybíjení, proměnná Vybíjení ma hodnotu jedna, nastaví se PoruchaPamet na hodnotu jedna. Je-li systém ve stavu nabíjení, nebo neaktivním stavu, nastaví se pouze varování.

Ventilátory mají kontrolu přetížení bimetalovými snímači, které jsou zapojeny do série. Dojde-li k přetížení některého ventilátoru, bimetal se zahřeje a rozpojí obvod a tím se nastaví proměnná ChodVentilatoruChladice na nulu. Program tedy testuje tuto proměnnou, a když se nastaví na hodnotu nula, vypíše chybové hlášení a nastaví proměnnou Varovani na hodnotu jedna.

Varování

Dojde-li v systému k chybě, která není nebezpečná pro řízení systému, spustí se varování. Toto varování se spouští nastavením proměnné Varovani na hodnotu jedna. V tomto

stavu bliká kontrolka Porucha na rozvaděčové skříni. Stav této kontrolky se nastavuje proměnnou Porucha, která je zobrazována i ve vizualizaci. Pokud je proměnná PoruchaPamet nastavená na nulu, tak je při aktivaci varování v cyklech programu střídavě rozsvěcována a zhasínána kontrolka Porucha.

Porucha

Dojde-li v systému k chybě, která znemožňuje jeho řízení, aktivuje se stav porucha. V tomto stavu se rozsvítí kontrolka Porucha na rozvaděčové skříni, vypnou se obě čerpadla a všechny ventilátory a původní stavy se uloží do pomocných proměnných pro opětovné spuštění po kvitování poruchy.

Kvitování poruchy

V programu RychlaSmycka se při stisku tlačítka Kvitování poruchy nastaví proměnná KvitovaniPoruchyPomocna na hodnotu jedna. V programu Porucha je tato proměnná testována, a pokud je nastavena na hodnotu jedna, uvedou se čerpadla a ventilátory do stavu před nastavením poruchového stavu přiřazením ovládacím proměnným hodnoty z pomocných proměnných. Dále se provede vynulování proměnných KvitovaniPoruchyPomocna, Porucha, PoruchaPamet, PoruchaVentilatoru, TlakMinimalniNabijeciPomocna, TlakMinimalniChlazeníPomocna a Varovani. Pomocí cyklu LOOP se provede vynulování polí PovolitZapisChyby a PovolitZapisChybyPam. Proměnná Regulace se nastaví na hodnotu jedna.

6 Testování

Testování jednotlivých částí probíhalo současně s programováním. První programovanou částí byla komunikace kalorimetru. Byla naprogramována základní struktura programu pro odeslání požadavku na kalorimetr a následné přijetí odpovědi. V té době však ještě nebyl k dispozici kalorimetr, aby byl program otestován, proto byla programována další část programu a to regulace zpátečky. Po naprogramování této části byla funkce ověřena a vše fungovalo podle požadavku.

Dalším krokem v programování byla regulace čerpadla. Při jeho zprovoznování bylo zjištěno, že projektant nevěděl, že se musí po zapnutí napájení frekvenčního měniče počkat 20 sekund a poté aktivovat vstup měniče pro rozběh motoru. Byl tedy napojen výstup z PLC pro spuštění motoru. První verze programu pro řízení čerpadla byla vytvořena jako přírůstkový regulátor. Při testování se regulátor neosvědčil. Měl velmi dlouhý přechodový děj. Proto byl program předělán na regulátor PS. Proporcionální a sumační konstanty byly určeny experimentálně. Při testování programu vznikaly jevy, kdy se čerpadlo náhle zastavilo a opět se začalo rozbíhat. Při analýze problému bylo zjištěno, že dochází k přetečení sumační proměnné a tím k náhlé záporné akční veličině. Byl tedy odstraněn problém s přetékáním a regulace čerpadla začala fungovat podle předpokladů. K této části bylo doprogramováno hlídání funkce čerpadla, které fungovalo podle představ.

V průběhu programování a testování regulace čerpadla, byl dodán a namontován kalorimetr. Provedlo se napojení kalorimetru na rozhraní RS485 a začalo testování. Po odeslání požadavku na kalorimetr neodesílal odpověď. Posléze bylo zjištěno, že modul IF1030 je defaultně nastaven na rozhraní RS422. Po změně nastavení rozhraní začala komunikace fungovat. Dále bylo programováno sestavení proměnných z příchozího bajtového rámce. Nakonec bylo experimentálně zjištěno, jakým způsobem se provádí kontrolní součet a tak bylo do programu zavedeno kontrolování kontrolního součtu.

V průběhu programování všech částí byly do části programu Porucha přidávány potřebné operace, které se mají provést při vzniku poruchy. Do této části byla nakonec doprogramována rutina, obsluhující poruchy teplotních čidel, a do programu ExportData bylo přidáno vytváření a zápis do souboru s chybovými hláškami.

V průběhu programování bylo vytvořeno řízení ventilátorů chladičů. Tuto část nebylo možné dosud otestovat. v současné době je systém v režimu nabíjení a až posléze bude probíhat režim vybíjení, kdy bude možné otestovat řízení ventilátorů.

Zadavatelem bylo zjištěno, že ve vizualizaci není zobrazován skutečný stav ventilátorů, zda běží nebo ne, ale stav je zobrazován na základě stavu výstupu automatu. V současné době je v řešení, aby byly dovedeny signály ze stykačů a tím mohl být zobrazován skutečný stav, zda jsou ventilátory zapnuty, nebo nikoli.

7 Závěr

Po zhodnocení dosavadního stavu měření teplot v geotermálních výměnících tepla pro tepelná čerpadla byl navržen monitorovací systém na bázi PLC B&R X20 od rakouské firmy Bernecker&Rainer. Pro měření teplot byly vybrány dva druhy teplotních čidel. Pro měření na vstupech a výstupech energetických vrtů a v systému nabíjení a vybíjení byla vhodnější odporová čidla PT1000, připojena k analogovým zásuvným modulům automatu. Pro měření v pozorovacích vrtech byla vhodnější digitální čidla DALAS. Tato teplotní čidla komunikují s automatem pomocí linky 1-Wire, čímž se značně ušetří na kabeláži. Protože automat nepodporuje komunikační rozhraní 1-Wire, byla použita převodníky tohoto rozhraní na RS232.

Monitorovací systém byl sestaven a namontován do rozvaděče. Pak bylo provedeno napojení na silnoproudou část, rozvedení teplotních čidel a jejich napojení. Dále byl vytvořen program pro automat, komunikující s kalorimetrem a teplotními čidly DALAS a řídicí režim nabíjení a vybíjení. Automat odesílá naměřená data po síti Ethernet na server umístěný v areálu Vysoké školy báňské, ale zároveň provádí zálohu naměřených dat na FTP server v automatu. v případě výpadku síťové komunikace doplní správce serveru chybějící data ze zálohy v automatu.

Při programování jednotlivých částí byly rovnou prováděny testy naprogramovaných částí. Základní funkce je tedy otestována. V současné době je zásobník v režimu nabíjení. Zatím se neprojeví žádné chyby v řízení. Nebylo ještě možné otestovat režim vybíjení, který bude spuštěn po nabití zásobníku.

Použitá literatura

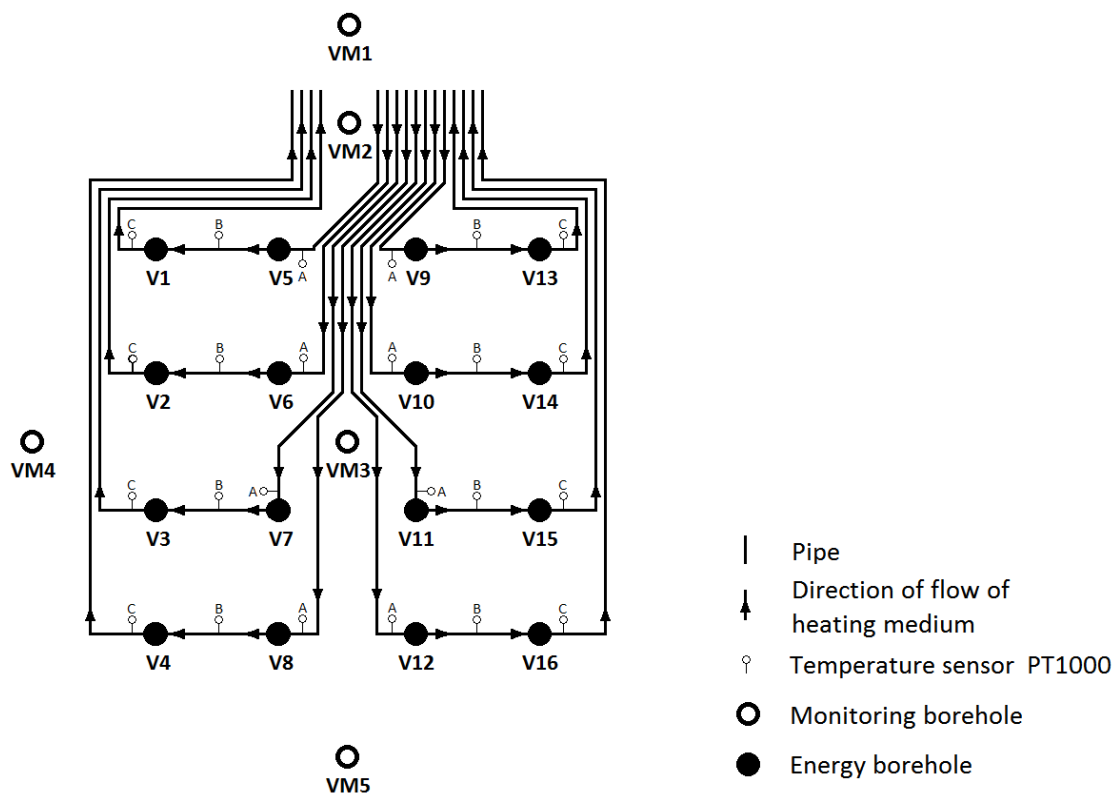
- [1] KOZIOREK, J., BUJOK, P., VRTEK, M., HÁJOVSKÝ, R., VOJČINÁK, P., LÁTAL, J., KLEMPA, M. *Studie metodiky měření vlastností horninového masivu v návaznosti na efektivní provoz tepelných čerpadel* [online]. Ostrava, 2011 [cited 27 April 2012]. Available from: X20 System User's Manual [online]. 2.10th ed. 5/2009 [cited 14 April 2012]. Available from: http://www.sensit.cz/ke-stazeni/soubory/Pt1000_3850.pdf.
- [2] *X20 System User's Manual* [online]. 2.10th ed. 5/2009 [cited 14 April 2012]. Available from: http://www.sensit.cz/ke-stazeni/soubory/Pt1000_3850.pdf. [3] BRZEZINA, P. *Čidla teploty PT1000* [online]. Ath ed. [cited 02 April 2012]. Available from: http://www.sensit.cz/ke-stazeni/soubory/Pt1000_3850.pdf.
- [4] VAVRLA, T. *Měření teplot pomocí programovatelného automatu B&R a teplotních čidel připojených na sběrnici 1-Wire: diplomová práce*. Vysoká škola báňská - Technická univerzita Ostrava, 6.5.2011.
- [5] COMAC CAL Montážní a technické podmínky. *CALOR 40*,
- [6] RS-485. <http://cs.wikipedia.org/wiki/RS485> (accessed April 06, 2012).
- [7] M-Bus. <http://cs.wikipedia.org/wiki/M-Bus> (accessed April 06, 2012).
- [8] COMAC CAL M-Bus communication protocol specification. *CALOR 40*,

Přílohy

- **I. - Calor 40 M-Bus** (v pdf formátu na přiloženém CD)
- **II. - Dokumentace monitorovacího systému** (v pdf formátu na přiloženém CD – neveřejná část)
- **III. - Rozšířený abstrakt v anglickém jazyce**
- **IV. - Ovladač pro komunikaci s kalorimetrem** (neveřejná část)
- **V. – Program pro PLC** (na přiloženém CD – neveřejná část)

Příloha č. III. – Extended abstract

The subject of the diploma thesis is to design, program and assemble a monitoring system for an underground heat accumulator. The system is designed for Green Gas DPB, a.s., company, which is based in Paskov. The company is performing a research project, partly granted from the European Union, which is supposed to test the possibility of storing the heat energy in the Earth's crust and subsequently using it for heating. The company has developed a system of 16 energetic and 5 monitoring boreholes in an open operating hall. The energetic boreholes are spaced equally on a square area and are heated by a heat circuit with a gas cogeneration unit. Monitoring boreholes are supposed to monitor the temperature profile of the crust. One of them is located in the middle of the accumulator and rest is located around it.



The task of the monitoring system is to measure the temperatures on the inputs and outputs of the energetic boreholes and also the temperatures at required depths in the monitoring boreholes. The temperatures are sent to and stored at a server located at VŠB. The data will be used for further research and evaluation. The system should also be able to read and send the actual data from the calorimeter. Another task of the monitoring system is to provide a simple regulation of charging and discharging of the accumulator and to watch over possible failures. The subjects of the regulation are a pump controlled by a frequency inverter, the circulation pump of the cooling circuit, ventilators of the coolers and regulator of the temperature of the heating medium returning to the cogeneration unit.

When developing the design of the system, previous experience with measuring the temperatures in geothermal exchangers for heat pumps, which had been performed in the VŠB-

TUO grounds, was evaluated. There are two drilling polygons in the grounds, Large Research Polygon (LRP) and Small Research Polygon (SRP), each of them using different system for measuring and archiving. System PAC, product of ICPDAS company used on LRP, measures the temperatures with PT1000 temperature sensors and sends them to the server through Ethernet. However, such implementation has turned out to be insufficient, because communication failures lead to loss of the measured data, and also it is impossible to create a control system in a simple way. On the other hand, measurement system controlled by PLC X20 (Bernecker&Rainer product) is used on SRP. This system backs up measured data on its own FTP server, so the data lost during a communication failure can be restored. One disadvantage, shared by both of the systems, is that the conductors of the PT1000 temperature sensors are not brought directly into the evaluation unit. The connections are therefore subject to corrosion, resulting in more frequent failures. The other disadvantage is that four conductors are needed to lead from each sensor.

PT1000 temperature sensors were originally designed for all the measuring points. During the realization the temperature sensors in monitoring boreholes were changed from PT1000 to DALAS. These temperature sensors communicate with the monitoring device on the 1-Wire line. One wire is introduced into each monitoring borehole and the temperature sensors are installed on the wire in appropriate distances. Bernecker&Rainer's Programmable Logic Controller X20 is used for monitoring and control.

When the work on this diploma thesis started, the realization of the system was at the very beginning. Just the deep boreholes were created and equipped with U-tubes for heating. Therefore it was not possible to go through the system and examine it to make one's own idea. Also no detailed description on the functioning of the system existed, so frequent communication with the sponsor's supervisor was essential.

The compilation of the monitoring system according to the project documentation and its installation into service formed the practical part of the thesis. The connection of inputs and outputs was performed, the desk was attached to the power supply network, and the temperature sensors were distributed and connected to the measuring modules. The next step of the practical development of the thesis was programming the control program of the PLC which would be able to administer the communication with the calorimeter, the regulation and the evaluation of failure states.

The calorimeter measures energy, power, flow rate, volume and input, output and differential temperatures. The calorimeter is attached to the PLC by the RS485 interface, which was evaluated as the most suitable among the calorimeter's interfaces that are in the offer. Communication protocol M-Bus is used for communication. This protocol is based on the Master/Slave model of communication where byte frames are sent between the devices. The PLC (Master) sends a byte frame to the calorimeter (Slave). The byte frame contains the address of the calorimeter and a request for sending back the actual measured data. The calorimeter accepts the request and sends a byte frame with the requested data back to the PLC. The byte frame has a clearly defined structure. Therefore, according to the structure, the values of the individual variables can be compiled: all the measured values have a length of four bytes that are put together into the individual variables according to their data type. The byte frame includes one byte with a control sum. This value was neither further specified in the

documentation to the calorimeter nor the producer responded to an email enquiry about it, so it was necessary to carry out some experiments to find more about the way how the control sum is generated.

During the subsequent system testing it was found that the calorimeter is cannot operate with negative temperature differences, and therefore is not able to measure discharged energy. Consequently, a modification of the program of the calorimeter was requested from the manufacturer. The manufacturer added a new variable that measures the energy withdrawn from the accumulator. A modification of the PLC program had to be made, because by adding a new variable to the byte frame the whole frame shifted and the rules for the compilation of the variables changed. Also a program for calculating the total energy (by subtracting the discharged energy from the charged energy) had to be created.

Another part of the program was controlling the temperature of the returning medium. After the controller is turned on, the three-way valve is closed and the management is started, which is done by simple comparison of the actual temperature of the medium returning to the cogeneration unit with a preset value.

The regulation of the flow is done by a PS regulator with a feedback from the calorimeter, which controls the pump. The function of the pump is regulated by comparing the requested and actual rates of the flow. If the difference is bigger than 20% of the requested flow for more than a minute, the pump is restarted. After three unsuccessful attempts to achieve the requested flow a failure state is announced.

The monitoring of the failure states is the next task of the program. There are two categories of failures. Small failures, when there is no risk of emergency status of the system, are reported only by a warning signal (flickering of the indicator “Porucha”). When a serious failure is announced (there is a risk of emergency), an overall system shutdown is carried out and the failure is reported by the indicator “Porucha” lighting.

The measured data and all the system states are sent to a server located at VŠB-TUO, where the data are stored in one-minute intervals. The data are used for the internet visualization and further processing and evaluation. The data and system states are also backed up in five-minute intervals at the FTP server of the PLC in case of the communication failure.

Testing of the individual parts of the system was carried out simultaneously with programming. At present, the whole system is subject to a complex testing under normal operation. Some states of the system could not yet be tested – for example the discharging mode, which will be run after the accumulator is fully charged.